Using Discuss

Stan Zanarotti
Ken Raeburn
Nancy Gilman
MIT SIPB
Revision 1
11 April 1987

Copyright © 1987 Student Information Processing Board
of the Massachusetts Institute of Technology

1 What is Discuss?

Discuss is a networked, electronic conferencing system. It is similar to bulletin-boards that have become popular on microcomputers, except that it is accessible from time-sharing machines and workstations on the network. Discuss allows a user to attend a set of electronic meetings. Each meeting has a specific focus, such as a project, class, or special-interest.

Users may select ("attend") meetings, and read messages ("transactions") that have been entered in those meetings. A user may choose to enter new transactions into a meeting, either by replying to a previous one or starting one on a new topic.

2 History of Discuss.

Discuss is a project started by Bill Sommerfeld, Ken Raeburn and Stan Zanarotti of the Student Information Processing Board (SIPB) at MIT. It grew out of an interest in developing a networked version of Forum, the electronic conferencing system on Multics.[1] Stan had also developed a previous conferencing system on VAX/VMS that was named DISCUSS. Discuss uses a lot of ideas from Multics Forum, and a Multics Forum user would be comfortable with the user interface available for Discuss.

Discuss has received generous support from three departments at MIT: the Laboratory for Computer Science, Project Athena, and Information Systems. They allowed the above individuals to spend time on the project, in hopes of a usable system. Other SIPB members have contributed to Discuss, notably Robert French and Ted T’so.

3 Clients and servers.

Discuss uses a client/server model of networking. Discuss servers actually store the meetings, while the Discuss client programs handle the user interface. When a user runs one of the Discuss client programs, it contacts the various servers to manipulate the meetings. This division should be transparent to the user, but unfortunately nothing is ever totally transparent, especially in the event of failures.

There is currently only one client program available for general use (there are some special-purpose ones not of interest to the general user), but others can be written moderately easily. A client program based on the X window system is being developed and others should be on their way.

Because of the networked facet of this system, some sort of authentication scheme needs to be used. If such a scheme is not used, it becomes easy to generate fake electronic mail messages and enter them as transactions. In order to prevent fake transactions and to ensure privacy, the Kerberos system, developed on MIT’s Project Athena, is being used to authenticate user transactions. Although Kerberos is not completely impervious, it is the most secure method now available, even if it is not the simplest. If you are using an Athena workstation, there is nothing further you need to do to use Kerberos. Consequently there is nothing further you need to do to start using Discuss. If you are on an Athena timesharing system, in order to get Discuss to work correctly, run the "kinit" program, which will ask for your username and password; for reasons that would be better explained elsewhere, you should then run "kdestroy" before you log out. On Eddie and on most Athena timesharing machines, these programs can be found as "/krb/kinit" and "/krb/kdestroy" respectively.

4 Naming a meeting.

Discuss does not keep a central list of all Discuss meetings. Instead, it maintains a list for each user of the meetings that they attend. To name a meeting you can use the meeting name, known as the Meeting ID, or your own abbreviation. In the future it may be possible to use another user’s abbreviation by specifying the username in the command line. This approach allows users to have their own names for meetings. For example, the "Discuss_development" meeting could be known as "discuss" to one user, "dsc" to another, or by both to yet another user. All this information, as well as the information needed by the client programs, is stored in a file

[1] Multics Forum was originally written by Jay Pattin, J. Spencer Love, and Jeff Schiller, all SIPB members.

called ".meetings"[2] in your home directory. Without this file, Discuss will not be able to locate meetings. You can change the name and/or location of this file, and set the environment variable MEETINGS to indicate its new pathname.[3] This file is re-created and deleted by Discuss in the process of updating the information in it, so that a system crash or program interruption will not cause loss of data. It may cause an update in progress to be lost, but the data will be intact as one version or the other. This does mean, however, that you shouldn't set it up to refer to a symbolic link, because the link will simply be removed.

5 Using Discuss.

5.1 Initial setup.

Well, the best way to understand the ideas above is to start playing with the system. They should become clear after a little practice. Here is a step-by-step explanation of how to use Discuss.

To start with, run Discuss. The first thing it will do is to look for a .meetings file. If you do not have one it will assume this is your first time running the new version of Discuss. It will then ask you if you want it to create a .meetings file. Type "y". If this is your first time ever using Discuss it will create a prototype ".meetings" file in your home directory for you to start with. If you used the previous experimental version of Discuss it will convert your .disrc file to a .meetings file.

NOTE: On Athena workstations, Discuss software will be located in /usr/sipb/bin. On Athena's timesharing systems, the software is in /usr/unsupported/sipb. To run the system commands descried in this document (as opposed to the commands entered to the discuss: prompt), you will need to type "/usr/sipb/bin/discuss", "/usr/unsupported/sipb/discuss", etc. To avoid typing all this you can add /usr/sipb/bin (or /usr/unsupported/sipb) to the variable PATH in your .login file. If you don't know how to do this, see the SIPB's *Inessential Guide to Athena*.

Running Discuss without a .meetings file looks like this:

```
charon% discuss
discuss: No '.meetings' file

If you are using discuss for the first time, or if you have only used the
experimental version of discuss, you need to run the 'dsc_setup'
command from the shell.

Run dsc_setup now? (y or n)  y

Running setup...
  Creating .meetings file:
  done.
discuss:
```

The version of this program distributed on Athena will give you a .meetings file that lists two meetings, called "New_Meetings" and "Everybody". The former is for announcements of new meetings, and the latter is for random announcements, questions, and anything else that may be of general interest.

[2]This is actually relatively new to Discuss. The original intention was to have two names, one longer, more informational name, and a shorter name more easily entered on the keyboard. The current impression is that this scheme of allowing arbitrarily many names would make things more convenient for some users. However, as this is a very new feature, many commands still deal only with two names for a meeting. The important exception, "goto", will accept any name for a meeting. Worst of all, there are no commands to manipulate the list of names on a meeting yet. The file can be edited manually, though..

[3]By the way, if you like using RFS you can set MEETINGS to indicate a file on another machine

5.2 Finding interesting meetings.

To list the currently available meetings type "list_meetings" and RETURN. Without any arguments, "list_meetings" lists out the meetings that you attend, i.e., the meetings in your .meetings file.

```
discuss:  list_meetings
 Flags  Meeting ID                         Short name
        Everybody                          eve
        New_Meetings                       new_meetings
discuss:
```

You may use any of the names to attend a meeting. To attend the New_Meetings meeting, simply "goto" it. Here we use its short name, which is generally easier to type. (Well, okay, this one is not shorter, but it doesn't contain mixed cases.) The ".meetings" implementation allows for any number of names on a meeting; however, this is a relatively new feature, and it is not fully supported yet.[4]

```
discuss:  goto new_meetings
New_Meetings meeting;  27 new, 27 last.

discuss:
```

The transactions in "New_Meetings" can be used to add the meetings you wish to "attend" to your personal list of meetings. First, list all the transactions.

```
discuss:  list all
 [0001]* (2) Oct 29 21:54 wesommer        Reason for this meeting
 [0002]  (9) Oct 29 22:04 wesommer        Discuss_development meeting.
 [0003]  (7) Oct 29 22:07 wesommer        Panama meeting
 [0004]  (8) Oct 29 22:17 wesommer        Steve_Changes and Oliver_Changes
 [0005]  (8) Nov  4 15:52 wesommer        Rumours
 [0006]  (9) Nov 11 20:12 srz             eve
 [0007]  (5) Nov 17 18:59 ambar           assassin-flame
 [0008]  (8) Dec  7 04:10 spook           Installations meeting
 [0009]  (8) Jan 18 19:24 rfrench         Info-Amiga meeting
 [0010]  (8) Feb  5 13:18 srz@LCS.MIT.EDU X_shell meeting
 [0012]  (6) Feb 12 20:35 wesommer        Martian_SHARKS meeting
 [0014]  (5) Feb 16 16:31 ambar           Nemesis meeting
 [0015]  (5) Feb 20 16:06 jtkohl          Compact_Disc meeting
 [0019]  (5) Feb 21 18:52 wesommer        Notification_Service meeting
 [0020]  (6) Feb 22 22:21 ???             Central_America meeting
 [0021]  (9) Feb 26 13:48 wesommer        Athena_Meetings meeting
 [0022]  (9) Mar  4 11:03 tytso           quotes meeting
 [0023]  (9) Mar 14 12:22 ambar           TeXlists meeting
 [0024]  (5) Mar 18 21:04 rfrench         NutWorks meeting
 [0025]  (1) Mar 19 12:18 jtkohl          Re: NutWorks meeting
 [0026]  (5) Mar 20 15:12 rfrench         Re: NutWorks meeting
 [0027] (19) Mar 23 13:04 geer            File_system_model meeting
discuss:
```

The different columns in the listing show the number of each specified "transaction" in the meeting (with a "*" next to the "current" one), the length in lines, the date and time of entry of the transaction, the user who entered the transaction, and the title given to it. Transaction 10 is from "srz@LCS.MIT.EDU"; this means that Stan

[4] The names are kept in a comma-separated list in the .meetings file; only the first two are used by list_meetings, but any will work as a name for the meeting.

(whose username is "srz") was using Discuss from LCS when he entered it.[5] Note that some numbers are unused in the sequence; these transactions have been deleted, probably by the user who put them there. They are therefore invisible to you, unless retrieved (un-deleted, see section 5.4) by the writer of the transactions at some future point in time.

Transactions are specified by number. There are several ways to select a range of transactions. To specify the first ten messages you can type "1,10" or "1:10". Keywords ("all", "last", "first", "next", etc.) can be used to choose a specific transactions or a range. "last-1" selects the next to the last message and "last-10:last" selects the last 10 messages. For more information type "help spec" in Discuss.

To learn more about a meeting listed in "New_Meetings" "print" the corresponding transaction (This transaction is the same as the first transaction in the announced meeting). For instance, if "Panama" sounds interesting you would do the following:

```
discuss:  print 3
[0003] wesommer@ATHENA.MIT.EDU New_Meetings Oct 29 22:07:32 1986 (7 lines)
Subject: Panama meeting
  Meeting Name:  Panama
  Hostname:      CHARON.MIT.EDU
  Pathname:      /usr/spool/discuss/panama
  Participation: Public

This meeting is fed nightly with all the changed .plan files of users
on Charon.[6]
--[0003]--
^L
discuss:
```

The "hostname" and "pathname" headers are for the "add_meeting" routine to use; they tell Discuss where to find the meeting. The "participation" field pertains to access control; see the section on access control in this document for more information.

The ^L at the end of the transaction marks the of the message. If you are reading a range of transactions the ^L signals the print command (which uses "more") to stop and wait for you to press space bar before going on to the next transaction. If a transaction takes up more than one screen, print will put up the first screen and wait for you to press spacebar before going on the the next screen.

If you want to read the transactions in Panama you need to add it to your personal list of meetings. To do this use the command "add_meeting" ("am" for short).

```
discuss:  add_meeting
Transaction [0003] Meeting Panama (panama) added

discuss:
```

Without any arguments, "add_meeting" looks at the current transaction, meeting announcement, and adds that meeting to your private list of meetings. Now "list_meetings" (lsm for short) will show your new list of meetings:

---

[5]The Kerberos realm of the writer of the transaction is recorded (not the hostname), and is displayed by "list" when it differs from the current local Kerberos realm.

[6]Don’t laugh, it gets some neat stuff.

```
discuss:  lsm
 Flags  Meeting ID                         Short name
        Everybody                          eve
c       Panama                             panama
c       New_Meetings                       new_meetings
discuss:
```

The "c" under the "Flags" column indicates that the meeting has changed -- There are transactions which you have not read. The flag will not appear if you have never been to that meeting.

Another way to add a meeting is to get its location from somebody else on the local machine. If you know that user "wesommer" on your machine knows about a meeting "assassin-flame" that you want to get to, you can add it to your own list by giving the full meeting name to add_meeting:

```
discuss:  am assassin-flame -user wesommer

discuss (add_meeting): Assassin-flame meeting (assassin-flame) added.

discuss:
```

(Unfortunately, this doesn't work well in Athena's public workstation scheme, since other users won't have home directories or ".meetings" files on the public workstation you use, but we're working on it.)

5.3 Saying something.

There are two ways you can enter a new transaction in a meeting. One is as a response to a transaction entered by someone else. For this, use the "reply" command. This will start up an editor (using the environment variable EDITOR, or "/bin/ed" if EDITOR isn't set[7]). for you to enter the text, and after you have saved the text and exited the editor, it will be entered as a response to the "current" transaction (generally the last one you looked at). If you just want to put in a quick line or two, use "reply -no_editor", and no editor will be started. Instead, all input up to a control-D or a "." on a line by itself will be taken as the text of the message.

The other way of entering a transaction is to start a new chain of transactions and replies, with a new subject. The command to do this is "talk" ("t" for short), "enter", or "send".[8] The "subject" field (which is printed at the head of each message and is displayed with "list") will be asked of you, and then an editor will be started as with "reply".

5.4 Oops, I didn't mean to say that.

If you decide the transaction you just entered offends all bald-headed Norwegians, or for some other reason should not be there, it can be deleted. Only the owner of a transaction, or someone with "delete" access to the meeting (see the section on access control), can delete a transaction. The command "delete" will mark a transaction or set of transactions as deleted; it is effectively deleted.

Transactions can also be retrieved if they have been deleted, with the "retrieve" ("rt") command. "Delete" access is required to retrieve a transaction not owned by the retriever; if the retriever owns the transaction, "owner" access will suffice. (Again, see the section on access control.)

---

[7]Athena's default startup files set the EDITOR environment variable to "/usr/athena/emacs".

[8]The reasons behind having three different names mostly are historical; Multics Forum has the concept of an "unprocessed" transaction, one for which the text has been entered but the transaction has not been entered into a meeting. The "talk" command would start processing (input) of a new transaction, which was not necessarily entered. The "enter" or "send" command would enter the unprocessed transaction into the meeting. In Discuss, unprocessed transactions have not been implemented (at this point) , and probably will not be in the near future, if at all; there will probably be a sort of "whatnow" loop which you may have seen in OLC and some MH commands.

6 Access control.

6.1 Kerberos and Discuss

This document is not intended to go into the details of Kerberos. Although it is hoped it will explain what you need to do in order to use Discuss. Basically you need a Kerberos password and each time you login you need to be "authenticated" by getting a set of "tickets". The procedure for this is explained below.

- On Athena, you have a Kerberos password if you have changed your password on any timesharing machine since about mid-January. You can even enter your old password as your "new" one. If you are outside of Athena and "kinit" fails, contact your Kerberos administrator. At LCS, this is Stan Zanarotti (username "srz@grape-nehi.lcs.mit.edu"); at Eddie, this is Jean Diaz (username "ambar@eddie.mit.edu").

- If you log in to an Athena workstation to use Discuss, make sure that when you log in, you don't get a message "No tickets obtained." If you get this message, this probably means you do not have a Kerberos password and should get one.

- If you log in to an Athena timesharing system to use Discuss, run "/krb/kinit" before using Discuss (it will ask for your username and password), and "/krb/kdestroy" after you are finished. It's not a bad idea to put "/krb/kdestroy -f" in your ".logout" file. If you are working on Athena's staff machines, they have kinit and kdestroy installed in /usr/athena, so you can run them as "kinit" and "kdestroy".

If you do not get Kerberos tickets by doing one of the above, you will probably get error messages from Discuss mentioning "Kerberos" (especially "Can't read Kerberos ticket file"), and will be treated as an anonymous user by Discuss, except for meetings on your local machine. If you have Kerberos tickets you will be treated as a Kerberos-authenticated user.

6.2 Possible access modes

The access modes that can be granted to a user are the following:

- `A:` answer -- User can enter replies to existing transactions. Without "own" access, however, he will not be able to delete his transactions.

- `C:` chairman -- Allows the user to modify the access list; this grants the user (indirectly) any access modes he desires.

- `D:` delete -- User can delete and retrieve transactions.

- `O:` own -- User can read, delete and retrieve his own submissions to the meeting, i.e., those he entered.

- `R:` read -- User can read any (non-deleted) transaction in the meeting.

- `S:` status -- User can retrieve certain information about the meeting, such as the number of transactions or access control list.

- `W:` write -- User may start new transaction chains in the meeting. Without "own" access, however, he will not be able to delete his transactions.

6.3 Public meetings vs Private meetings

The private and public flags in the participation header has no effect on the access control, but simply states whether the announcer considers the meeting to be public or private in nature.

6.4 Chairman access
The chairman or chairmen of a meeting can modify the access control list; they therefore can decide who can do what within the meeting. Chairman access does not automatically bypass the other access control terms; thus a chairman can give himself "acorsw" access (add, chairman, own, read, status, and write, but not delete) to prevent himself from accidentally deleting a transaction, though he can explicitly change the access to permit himself to do so.

6.5 Anonymous transactions.
Sometimes a transaction will be entered as being from user "???". This means that the authentication (discussed above) failed; the Discuss server had no proof of the user's identity. The "???" userid can appear in the access control list, and can be given restricted access; thus, a meeting can be set up to allow anonymous reading, but require authentication for entering transactions.

7 Some existing meetings.
There are several meetings which have been running for quite some time; some of them are listed below. Many of these are set up to be recipients of one or more mailing lists. If you want to read these lists you can use Discuss and save disk space. These lists are generally set up as read-only; if you want to reply, send to the mailing list(s).

- Discuss_development, discuss -- For discussion (pun intended) of development of Discuss, mention of bugs discovered, changes introduced.
- Panama, panama -- Started long ago on Charon, each transaction includes all changed ".plan" files since the previous transaction. Some of these start some interesting discussions (pun not intended).
- Everybody, eve -- For random flaming, discussions.
- Info-Amiga, amiga -- Repository for the INFO-AMIGA list from Rutgers. Digestified version of comp.sys.amiga (net.micro.amiga).
- Compact_Disc, cd -- For people who like compact discs and want to talk about them. I can't afford many, so I don't bother reading this. Someone want to give me a description?
- Central_America, ca -- Like Panama, but bigger. Checks plan files Athena-wide. Still in experimental stage, but interesting.
- TeXlists, tex -- Receives some TeX-related mailing lists.
- bboard -- Receives MIT bboard lists. Announcements, commands, lost+found, for-sale, etc.
- Athena_MSGS, msgs -- Receives messages to "msgs" on Athena.

8 More help

8.1 Summary of Useful Commands
The following are a list of common Discuss commands, and their abbreviations as well as a short description of each. It is not necessarily complete.

- help - gives information about the available commands (you need to give it a subject, see below)
- list_requests (lr, ?) - lists available commands
- quit (exit, q) - exits out of Discuss
- list_meetings (lsm) - gives you a list of the meetings you attend and indicates wether or not there are new transactions.

- goto (go, g) - allows you to go to (i.e. read/enter/respond-to transactions) a meeting (from your .meetings file).
- add_meeting (am) - allows you to add a meeting to your .meetings file.
- delete_meeting (del_mtg, dm) - deletes a meeting from your .meetings file
- announce_meeting (ann-mtg, anm) - announces a new meeting
- check_meetings (ckm) - gives you a list of the meetings in your .meetings file which have new transactions.
- next_meeting (nm) - goes to the next meeting in your .meetings file that has new transactions.
- print (pr,p) - prints transaction(s)
- next - prints the next transaction
- prev - prints the previous transactions
- talk (enter, send, t) - enters a transaction
- reply (rp) - replies to a transaction
- list (ls) - lists the transactions in a meeting
- write (w) - writes a transaction into a file
- delete (dl, d) - allows you to delete a transaction (as long as you have access).
- retrieve (rt) - undeletes a message (as long as you have access)
- list_acl (la) - lists the access control list for a meeting

8.2 Help files

While in Discuss, there is a "help" facility available. There is the equivalent of a "man" page for each command within Discuss, and also for "access" and "specifiers". You can type "help foo" for information on command or topic "foo".

Unfortunately, many of these are out of date...

9 Known bugs.

These are some currently known problems that have not been fixed yet:

- Not all commands handle ^C interrupts properly; sometimes you will get a "discuss:" prompt, sometimes you will break out of Discuss back to the shell.
- If a host is down or unreachable, Discuss will not remember this information; instead, it will try to open a connection to the host every time a meeting on that host is referenced.
- Broken connections are fatal to Discuss. If a host goes down, or if the remote server dies, Discuss will die with a "broken pipe" error. This will also happen if there is a meeting on your local host, and the subprocess server is not in the expected place.

9.1 Who to gripe to...

If you have complaints, bug reports, suggestions, or any other comments concerning the workings of Discuss, please enter them into the "Discuss_development" ("discuss") meeting, or send them via electronic mail to "discussers@athena.mit.edu".

Table of Contents