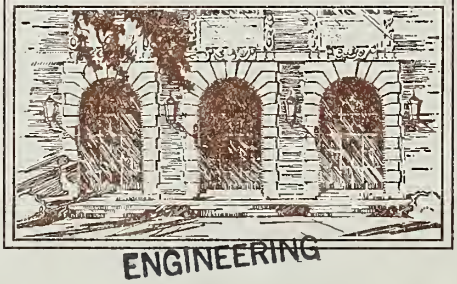


Center for Advanced Computation

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
URBANA, ILLINOIS 61801

CAC Document No. 89

Accounting System Primer

by

Peter A. Alsberg
and
John R. Mullen

Revised August 1974

CAC Document No. 89

Accounting System Primer

by

Peter A. Alsberg

and

John R. Mullen

Revised August 1974

Table of Contents

Table of Figures

Introduction

The Center for Advanced Computation accounting system was built by the Center for Advanced Computation to reduce the paper work in the business office and speed up the preparation of reports. The system knows about the University of Illinois accounting procedures and has been tuned to these procedures. It may not be acceptable for use in other environments.

The only program currently running in the system is a ledger program. The ledger program is of interest primarily to the accountant. At present it contains no sophisticated retrieval or analysis capabilities. It can only record transactions and write these transactions back out completely itemized and organized by account, by month, by year, and by expense class and subclass.

If sufficient interest is shown, other facilities will be added to the system to facilitate budget preparation, to sound alarms when expenses increase at a rate that jeopardizes budgets and to allow the retrieval of the full history of any purchase order over a multi-year period. Eventually, we expect that the system will maintain personnel files to help facilitate budget preparation and automatically log payroll transactions into the ledger system. We are also looking at a facility to reduce the amount of time spent rationalizing departmental records to the monthly university statement for each account.

The Computers Involved

The system runs on the Multics computing system at MIT in Cambridge, Massachusetts. We use Multics because

1. it is easily available from the Urbana campus;
2. it is the cheapest system for building this kind of accounting system;
3. we can build the complete system in a much shorter time than on any other computer;
4. Multics is the only system currently available that provides us with a high degree of security and privacy.

The link from the University of Illinois to the Multics computer at MIT is the ARPA computer network. All that the user normally sees is a typewriter-like terminal in his own office. A local computer at the Center for Advanced Computation, called ANTS (ARPA Network Terminal System), makes it easy to use the ARPA network and the accounting system at Multics. The accounting system user makes only one phone call and types 5 or 6 words. From that point on he is inside the accounting system at Multics and he can forget about networks and telephone lines and local computers like ANTS.

Ledger Program Overview

The ledger program treats each department separately. The departmental data contained in the accounting system can only be read and written by specifically authorized persons in each department. These data access permissions are controlled on a department by department and individual by individual basis. The programmers at the Center for Advanced Computation can show you how to set the various data access privileges.

Each department can choose its own expense classes independent of the expense classifications used in other departments. However, expense classifications apply to all accounts in the one department. All expense classes must be broken into expense subclasses. For example, a department might choose a simple three class scheme: *wages*, *equipment*, and *other*. Wages might be broken down into subclasses like: *faculty*, *professional*, *non-academic*, *undergraduate hourly*, *graduate research assistants*, etc. Similarly, the *equipment* class and the *other* class can be broken down into a variety of subclasses. Departments may have a very large number of accounts, classes, and subclasses if they choose. Departments may also specify multiple names and abbreviations for each of their accounts, classes, and subclasses.

Entering transactions into the ledger

The transactions in the ledger system are either temporary or frozen. When transactions are first entered into the system they are only temporary. While they are temporary you can change the values of some of the fields of the transaction, delete a transaction altogether or perform various checks and request sub-totals to check the validity of the temporary transactions. ONCE A TEMPORARY TRANSACTION HAS BEEN FROZEN IT CAN NEVER BE CHANGED AGAIN. The frozen transactions form a permanent record. These frozen transactions cannot be tampered with in any way. They cannot be intentionally or unintentionally altered or deleted.

Since frozen transactions are so permanent, special care must be taken to insure their accuracy. If, somehow, you still make a mistake and freeze an erroneous transaction, the only way to correct the error is to enter an offsetting transaction. A good working technique is to enter 10 or 20 transactions, check them for accuracy and freeze them. Then go on to the next 10 or 20 transactions. If for any reason either your terminal, your telephone line, ANTS, the ARPA network, or Multics should fail, do not worry. The accounting system is set up so that it remembers everything you did. The worst that can happen is that you might have to retype the very last line you were working on when your terminal or the telephone line or one of the various computers failed.

All input lines are terminated by a carriage return. It is possible to enter a textual description of each transaction. Sometimes this description may require more room than is available on a single line. This is no problem. You can continue descriptive information onto the next line and the next and the next after that. In fact, the system is set up so that you could write several hundred pages of description for one transaction if you really wanted to. To continue a textual description onto the next line on your terminal, type a plus, "+", as the first character on the next line. All continued lines will begin with a plus.

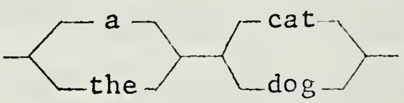
In order to enter a transaction you must specify the seven fields in the transaction. These fields are an action, a reference code, a dollar amount, a subclass, an account, a date, and the text field we just mentioned. Each of these fields is separated by one or more blanks. With the exception of the text field, none of the other fields may contain a blank. When you specify the names of subclasses, accounts, and classes you should be careful in your use of upper and lower case. The accounting system knows the difference between an upper case and a lower case letter. Each class, subclass, and account may have several names or synonyms. For example, you might have a subclass with the name Teaching_assistants. This subclass may also have synonyms TA, ta, and gradta. These additional names are for your convenience. The accounting system knows they all identify the same subclass and does not care which one you use.

Even with the heavy use of short synonyms it would still be a terrible bore to type in all seven fields all of the time. If you are entering a series of transactions and the last few fields of the next transaction are the same as the last transaction then you don't have to type in the last few fields. Type the first fields. When you get to the fields that are the same, type your carriage return to terminate the transaction. The accounting system will automatically copy the missing fields from the last transaction. You might also want to use the <u>ditto convention</u>. If the reference, dollar amount, subclass, account, or date field of the next transaction is the same as the last transaction you can simply enter a ditto mark (") in that field. The accounting system will automatically copy the dittoed field value from the previous transaction.

The format of a transaction is shown graphically in figure 1. The graph in figure 1 is called a <u>syntax diagram</u>. A syntax diagram describes all of the legal ways that a request can be formed. The rules for syntax diagrams are simple:

1. follow the path from left to right
2. where single or multiple characters appear, those characters must be put in the request in exactly the same way they appear in the syntax diagram
3. where phrases appear in pointy brackets, e.g. <subclass name>, they describe the kind of thing which must be put in the request at that point.

Syntax Diagram Examples:



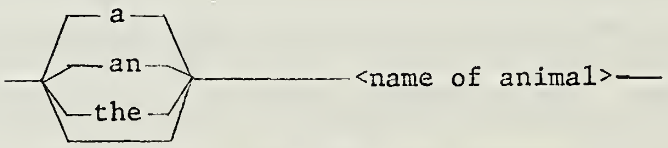
Means that the only legal requests are

a cat

the cat

a dog

the dog



has a very large number of phrases

a dog

an elephant

dog

cat

mouse

(note we took the bottom path in the first phrase and skipped the article altogether)

the mouse

etc.

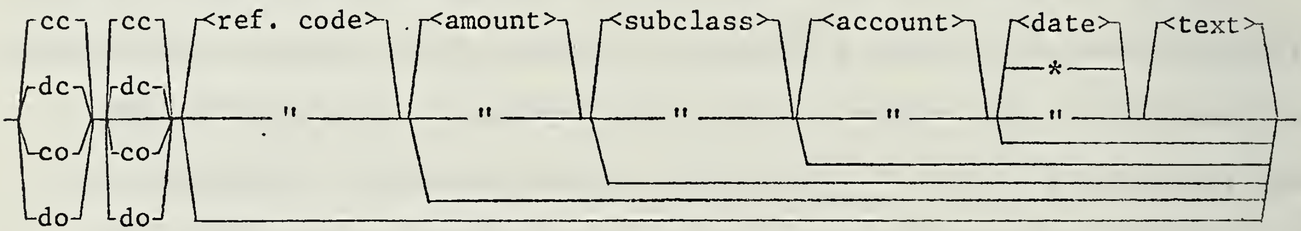
The ledger system has a very simple ledger format. There are only two columns of importance, the cash and the obligation columns. When you enter a transaction you must specify what actions are to be performed on the cash and obligation columns. You can either debit or credit one or both of the cash and obligation columns. For example, to have a transaction debit the cash column you would use the action code "dc". To have it credit the obligation column you would use the action code "co". You could specify that the transaction debit the cash column and credit the obligation column by using the action code "dcco". An action like dcco would commonly be used when recording an expense against a continuous purchase order.

The next field in the transaction is the reference code field. Normally, the reference code will contain a university purchase order number or a departmental requisition number. The reference code is very important. Eventually it will be used to provide a history of transactions so that you might, for example, go back many years and retreive all transactions that occurred relative to a given purchase order or requisition. The reference code must be at least one character long and may be up to eight characters long. Any character is valid except for the blank. You can feel free to use letters, digits, and marks of punctuation in a reference code. The third field of the transaction is the amount field. The amount to be debited or credited to the cash or obligation column is specified in dollars. No dollar sign should be typed in this field. Only digits and the decimal point are allowed. If an even dollar amount is specified, you do not have to use the decimal point. For example, $17.00 could be entered as 17 or 17.00 If the second digit after the decimal point is a 0 it may be omitted. For example, $17.20 could be entered as 17.2. No more than 2 digits are allowed after the decimal point. The maximum dollar amount that can be recorded in this system is approximately $343 million. Therefore, each transaction, total and subtotal must be less than 343 million dollars for each department.

The fourth and fifth field in the transactions specify a subclass name or synonym and an account name or synonym.

The sixth field in the transaction is the date field. The date must be written in month-day-year format with the month, day, and years specified as one or two digits and separated by hyphens or slashes. The accounting system always knows what day it is. If you specify an asterisk (*) in the date field it will simply use today's date.

The seventh field in the transaction is the text field. The text field is optional. It may contain blanks and it may be continued over many lines using the + convention.



cc = credit cash

dc = debit cash

co = credit obligation

do = debit obligation

dcco = debit cash and credit obligation

etc.

ref. code = 1 to 8 character reference code (examples: ABC 15x.3(L) 1347-L)

amount = amount to be debited or credited in dollars without the dollar sign (examples: 17 17. 17.00 17.2 17.23 14329.30)

subclass = any of the subclass names or synonyms established for the dept.

account = any of the account names or synonyms established for the dept.

date = date in mm-dd-yy format (examples: 8-13-73 10-3-72 2-38-75)

text = an arbitrarily long description of a transaction. If the first character on the next line is a "+" then the next line will be treated as a continuation of the text field. As many continuation lines as desired can be added to the text field by using additional lines beginning with a "+".

* = in a date, an asterisk means today's date

" = a ditto in a field means that the same field on the immediately previous transaction should be copied into the field of this transaction

Entering Transactions

figure 1

Examples of Transactions

The symbol (CR) is a carriage return.

1. add $15,700.00 to wages and subtract it from the equipment allocation for the NSF-Adams account

```
cc budget 15700 wages NSF-Adams  *  adjust budget (CR)
dc   "       "   equipment (CR)
```

The first transaction gave wages more cash. We made up the reference code "budget". The text field contains a short explanation of what was done. The asterisk specifies today's date. The second transaction subtracted the cash from the equipment subclass. The reference, amount, account and date fields are the same as the previous transaction so we only specified the action and subclass and used the ditto and carriage return conventions to automatically copy the 4 identical fields. The text field of the second transaction will be empty.

2. Obligate purchase order 1773-M for $100.37 of machine parts on account 44-77-36-735 (with alias 44/735)

```
do 1773-M 100.37  shop-equip  44/735 * misc. machine parts  (CR)
```

3. The machine parts of 2 came in but they only cost $98.73. Expense it and de-obligate it.

```
dc  1773-M 98.37 shop-equip 44/735 * price less than P.O. (CR)
co    "   100.37 (CR)
```

The first transaction shows that the equipment was paid for and how much it cost. The second transaction reduces the obligation column for that subclass by the amount originally obligated. If the delivered parts had cost the same as was obligated, the whole thing could have been done in one transaction:

```
dcco 1773-M  100.37 shop-equip  44/735  *
```

Checking and Freezing Temporary Transactions

Figure 2 is a syntax diagram which shows all of the legal requests used to process temporary transactions. All of the key words used when processing temporary transactions have a one letter synonym, their first letter. All requests to work with temporary transactions must be preceded by the word "temp" or its synonym "t". The purpose for this is two fold. The most important reason is that it will constantly reinforce, in your own mind, the difference between temporary and permanent transactions. The second reason is that the print and total commands will also work, in a slightly different fashion, on permanent transactions. By putting the t or temp before the print and total commands, the accounting system knows that you want to print or total temporary transactions as oppsed to permanent transactions.

Temporary transactions are kept in their own buffer. There can be up to approximately 25,000 temporary transactions in that buffer. Each transaction has a number assigned by the system. When you ask to have a transaction in the temporary buffer printed, the accounting system will also tell you what number it has assigned to the transaction. This number is used to reference the transaction whenever you want to edit it, delete it, print it, etc. There are 7 commands which can be used in the temporary buffer: count, print, total, set, replace, delete, and freeze.

The count command tells you how many temporary transactions are stored in the temporary buffer. If any temporary transactions have been deleted the count command will tell you how many.

The print command prints one or more transactions. When you ask for a print of a transaction, all of the fields in the transaction are reported to you. The transaction number assigned by the accounting system is also printed. The print command may have 0, 1, or 2 numbers following it. If there are no numbers, then it is assumed that the entire temporary buffer should be printed. If there is just one number, then the single transaction with that number will be printed. If two numbers follow the print command then the second number must be greater than the first number and all transactions from the first number up through and including the second number will be printed.

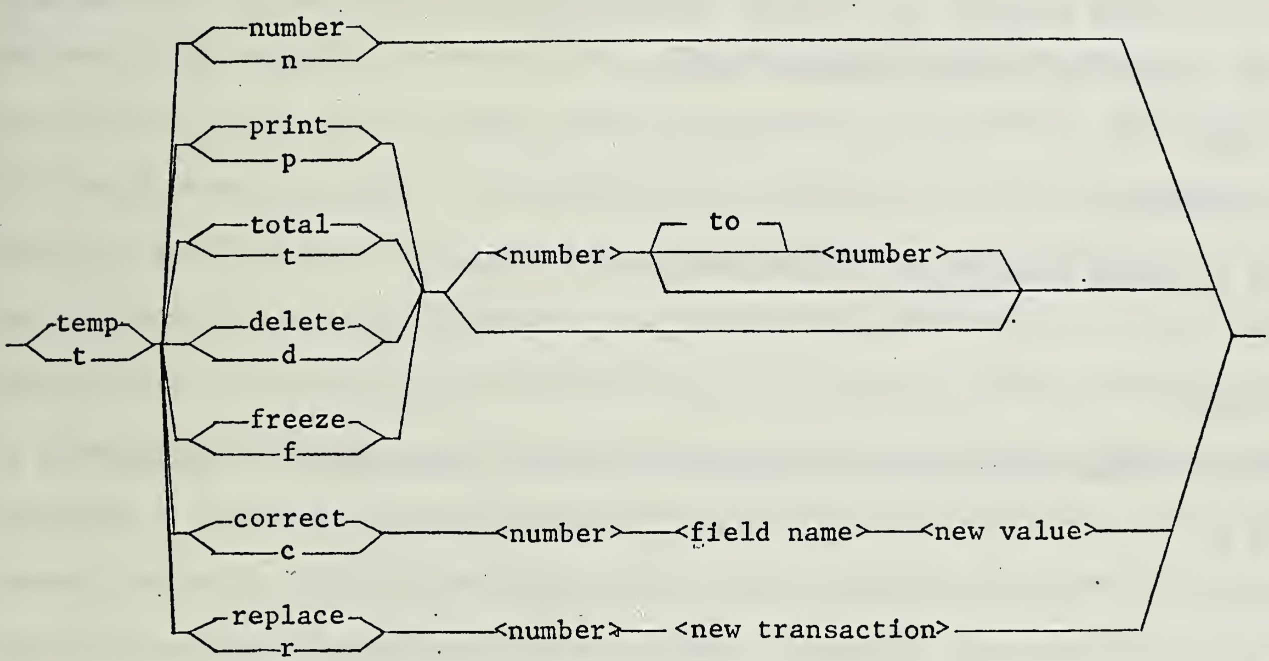
The total command produces an adding machine tape for temporary transactions. The total debits and credits to both the cash and obligation columns of the specified transactions is computed and typed out for you. As in the print command, the entire buffer, a single transaction, or a group of contiguous transactions may be totalled.

The set command allows you to correct a single field on a single transaction in temporary buffer. Following the set command is a transaction number, the field name, and the new value to be placed into the field. Figure 2 shows the allowed field names. Notice that most of the fields have short synonyms available for your convenience.

The replace command allows you to enter a new transaction and have it replace a single existing transaction. The replace command and the transaction number are typed and are followed by the new transaction. For your convenience the accounting system will assume that the last transaction entered was the transaction that was being replaced. Therefore, the ditto and carriage return conventions will copy fields automatically from the temporary transaction being replaced.

The delete command will delete an entire buffer, a single transaction, or a range of transactions. To protect you from your own typographical errors, the delete command will ask you if you are sure you want to delete the specified transactions. You must reply "yes" or "no" before the delete will be performed. If the accounting system decides that these are too many deleted transactions in the temporary buffer it will discard them and renumber the remaining temporary transactions.

The temporary freeze command will freeze all of the transactions in the buffer, a single transaction, or a group of transactions. Like the delete command, the freeze command will also ask you if you are sure you want to freeze the indicated transaction. Again, you must reply "yes" or "no". Like the delete command the freeze command may renumber the transactions. If this happens all frozen and deleted transactions in the buffer will be discarded. The remaining n transactions will be numbered sequentially from one to n in the order in which they were entered into the temporary buffer.



field names =	act action	ref reference code	amt amount dollar dollars	sub subclass class	acct account	date	text

Checking, Editing, and Freezing Temporary Transactions

figure 2

Examples of Temporary Transactions Requests

1. The following 4 requests will all print the number of active and deleted temporary transactions in the buffer:

```
temp count
t count
temp c
t c
```

2. The following commands will print all of the transactions in the temporary buffer:

```
t p
temp print
```

3. The following requests will produce an adding machine tape for the fifth through the 17th transaction in the temporary buffer:

```
t total 5 17
t t 5 thru 17
```

4. The following commands will freeze the entire buffer:

```
t f
t freeze
```

5. The following commands will be required to delete transactions 4, 7, and 9:

```
t d 4
t d 7
t d 9
```

6. The following commands will change the action and dollar amount fields of transaction 12 to "dc" and $100.00 respectively.

```
temp set  12 act dc
t s 12 amt 100
```

or

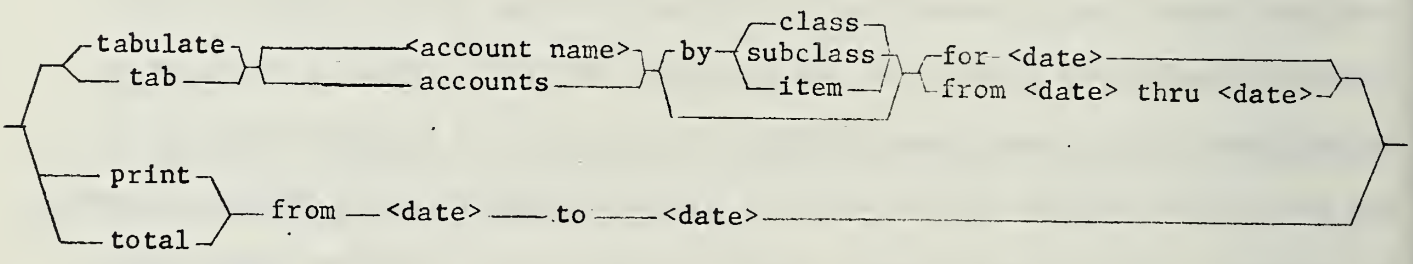
```
t r 12 dc " 100
```

Generating Reports on Permanent Transactions

There are three commands that work on permanent transactions. They are the tabulate, print, and total commands. Actually, you will only use the tabulate command and probably never use the print or total command.

The tabulate command, as currently implemented, will only print ledger sheets for all accounts in the department or for one account. In the future facilities will be added to allow you to prepare ledger sheets which give sub-totals for individual classes or subclasses only in one account or across all accounts in your department. Figure 3 is a syntax diagram for the tabulate request. The tabulate command has a synonym "tab". The second phrase in the tabulate request specifies whether all accounts in the department or just a single account are to be tabulated. The account name specified in the case of a single account can be any one of the synonyms for the account. The third phrase in the tabulate command is optional. If it is not present, then the default is an itemized ledger sheet. This third phrase specifies in what detail you would like the ledger prepared. The least detailed would be by class. If you specified by subclass then both class and subclass totals would be prepared and printed for each account. If you specified by item, then the ledger would have a class grouping. Within each class would be printed each subclass total and within each subclass would be printed an itemized list of all those transactions which occurred in that subclass for the specified account and month. The optimal phrase "with text" indicates that the text associated with each itemized transaction will be printed. Normally, the text will not be printed. The last phrase specifies the date. The date is given in the standard month, day, year format with the month, day, and year being one or two digit numbers separated by a hyphen. The day in the date field is ignored. The tabulate command is only concerned about the month. Tabulate will automatically tabulate all information for a given month. If this last field has the form "from <date to date>" then the two months indicated and all the months in between will be separately tabulated on the ledger sheets.

The print and total commands perform the same function on permanent transacactions that the temporary print and the temporary total commands perform on temporary transactions. Instead of specifying a range of temporary transaction numbers, a range of dates is specified. These dates must be in the same month. If only one date is specified, then all the transactions for all the accounts, classes, and subclasses, that were made on that day, will be printed in the order in which they were made. If two dates are specified then similar action is taken for all dates from the first date through the last date inclusive.



Report Generation

figure 3

Examples of reports on permanent transactions

1. to prepare a ledger for account 46/733 with expenses aggregated to the class level for the fiscal year-to-date (assuming it is October 21, 1973) type

```
tab 46/733 by class from 7-1-73 to 10-1-73
```

2. to prepare a ledger of all the accounts in the department, completely itemized, sorted and subtotaled by subclass and class for October, 1973 type

```
tab accounts by item for 10-1-73
```

3. to include the text associated with the transactions for the request above type

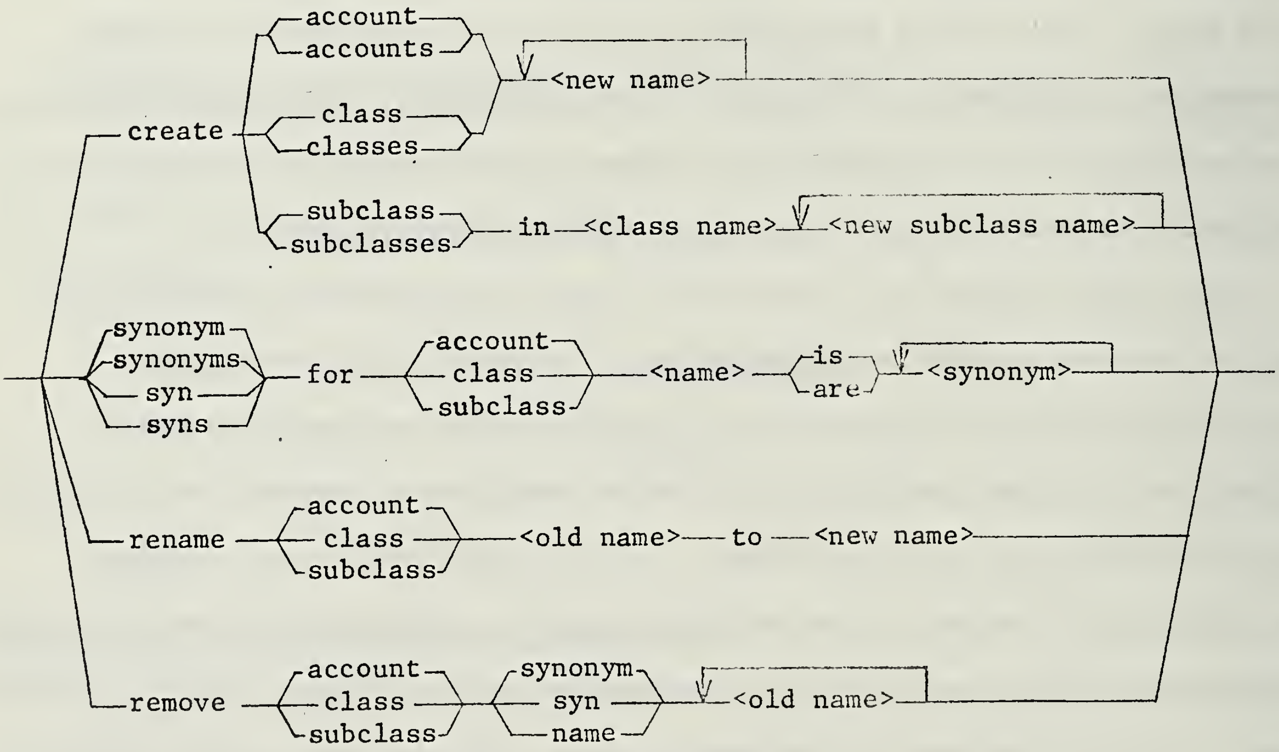
```
tab accounts by item with text for 10-1-73
```

Adding Accounts, Classes, Subclasses, and Synonyms

Accounts, classes, subclasses, and synonyms can be added freely while sitting at your terminal. Requests to add new accounts, etc., can be interspersed with the entry of transactions or any other command. Once created, an account, class, or subclass can never be destroyed. However, it can be renamed and synonyms can be added and deleted freely. Thus, you never need fear a mistake when adding a synonym or when renaming an existing account, class, or subclass. These actions can always be reversed. However, once created, an account, class, or subclass is with you forever. Any letter, digit or punctuation mark except a blank can be used in a name. Account names and synonyms can be up to 32 characters in length. For classes and subclasses the limits are 20 and 16 characters respectively. Figure 4 is a syntax diagram that illustrates those commands which create accounts, classes, and subclasses and which can add, change, and delete existing names.

The create command will create one or more new accounts or classes or one or more new subclasses within an existing class. The names and synonyms given to accounts, classes and subclasses must all be unique within each set of accounts, within each set of classes and within each set of subclasses. However, it is possible for you to give the name w to an account, w to a class and w to a subclass without confusion on the part of the accounting system. Because of the way you specify requests, the accounting system always knows whether you are talking about the account w, the class w, or the subclass w. Whenever a class is created with a given name, a subclass with the identical name is automatically created. The name given to each account, class, or subclass in a create statement will be the principal name of that account, class, or subclass. The principal name is the name that will be printed on ledger sheets and by the print command for both temporary and permanent transactions. Therefore, you should choose those names to be as complete and as descriptive as possible.

The synonym command adds additional names to any single account, class, or subclass. You can add an arbitrary number of synonyms to any existing name. The rename command will change the principal name or any synonym that currently exist for an account, a class, or subclass. The remove command will remove synonyms from any account, class, or subclass. If the principal name of an account, class, or subclass is removed, then the longest synonym automatically becomes the principal name. If there are no synonyms, then the accounting system will not allow you to remove the principal name.



maximum length of names:

account names ≤ 32 characters
class names ≤ 20 characters
subclass names ≤ 16 characters

names may contain any letter, digit, punctuation, or other character except the blank

Creating and Naming Accounts, Classes and Subclasses

figure 4

Examples of Name Modification

1. Add two new classes named wages and miscellaneous

```
create classes wages miscellaneous
```

2. Add the synonyms w to wages and m and misc to miscellaneous

```
synonyms for class miscellaneous are m misc

syn for class wages is w
```

3. Add the subclasses hourly, GRA, and faculty to the class wages

```
create subclasses in wages hourly GRA faculty
```

4. Rename the GRA subclass to be graduate-student

```
rename subclass GRA to graduate-student
```

5. Delete the synonym m from miscellaneous

```
remove subclass name m
```

Miscellaneous Commands

The *quit* command exits the accounting system and returns to the Multics supervisor. At this point you will most likely want to type the "logout" command. The logout command will then log you out of Multics and automatically break your ARPA Network connection to Multics.

The *help* command will give you information about the accounting system. To get help with any command simply type "help <command name>". Typing "help" without a command name will show a list of what help is available. For example, if you don't remember what the tabulate command syntax is, type

```
help tabulate  (CR).
```

Security

Security has been built into the accounting system from the ground up. It is not an add on. It exists in the first version. Security is one of the principal reasons we use Multics. No other computer system would let us build as secure a data system.

Security is under the control of the individual department. In particular, it is not under the control of the CAC.

All security systems can be broken. The degree of security is measured as the dollar cost to break security (e.g. through sophisticated equipment, bribes, etc.). The security that your accounting data enjoys in this accounting system should be the same as or better than it enjoys in your current system of open offices and locked file drawers. In the following paragraphs we will briefly discuss the nature of the security system and the ways in which it can be broken.

Each department has its own Multics project. Each project has a project administrator. The project administrator can specify who is allowed to use the data recorded for that project and the manner in which he is allowed to use it. Each person who uses Multics has a registered name on Multics. This name is normally his last name. If there is already someone registered on the system with his last name, then his first and possibly his middle initials are included before his last name so that he can be differentiated. No two persons have the same registered name. Each person, also, has a password which is linked to his registered name. The password is chosen by the user and can be changed at will. No one but the individual person knows his password. This information is not available to the project administrator or even to the Multics system administrator. It is extremely important that you never let anyone know your password or use a terminal logged in under your name. If it is important that someone have access to the accounting system data, then he should be registered on Multics with his own name and his own password. He should have his various data access permissions set explicitly by the project administrator. It only takes a few minutes to register a new user and the whole process is normally handled over the telephone.

Each piece of data in the accounting system has an access control list. The access control list describes which registered Multics users have permission to read the data and which registered users have permission to update the data. The access control list is controlled by the project administrator - not by CAC programmers. If he desires, the project administrator at the department could grant access to CAC programmers. He might do this if he needed special help. Once he no longer needed the help he could immediately delete the permission to access the data.

There are 5 basic ways the security system could be broken:

1. The project administrator at the department or one of the registered Multics users of the department who has permission to access departmental accounting data might release his password to some other person. New personnel should have constantly stressed to them the importance of keeping their password a secret. Even the project administrator has no need to ever know the password of one of his workers. If a password is requested by the project administrator, the password should not be given and the request should be viewed with suspicion.
2. The project administrator might intentionally sabotage the security system. He has the power to grant both read and update access to unauthorized personnel. The project administrator should be a trusted departmental employee-possibly the business manager.
3. The Multics system administrator at MIT could collaborate with unauthorized personnel and grant them read or update permission. It is a laborious but straightforward task for the system administrator to do this. It would not be in the best interest of the system administrator to collaborate in such a venture.

Quite to the contrary it could be extremely detrimental to his own interest. Unless he is extremely careful, he can leave behind obvious signs of his tampering. Furthermore, the tampering can be detected while it is in progress.

4. The project administrator and one CAC accounting system programmer could collaborate so that they could tamper with permanent accounting system records. The programmer normally has no interest in tampering with permanent records unless bribed. Furthermore, this kind of tampering leaves easily detected signs that cannot be hidden. Should a tampered data file be discovered, the original data file can be readily restored in its original form using the automatic Multics back-up procedures. Tampering with the back-up system would have to be done at MIT and would require the physical destruction of many reels of magnetic tape.
5. All of the CAC programmers could conspire together to introduce a "trojan horse" into the accounting system. In this case the accounting system is the trojan horse and contains within it a small piece of program which violates security. Whenever the project administrator or some other authorized person runs the accounting system, this small piece of illegal code is executed and might surreptitiously write confidential data into the files of another project. Once the data has left the control of the original department it may be read by unauthorized personnel. The trojan horse attack could be recognized by a programmer familiar with the accounting system program. Alternatively, a careful and lengthy examination of the accounting system program by an external programmer could reveal the trojan horse.

Security can be violated if departmental personnel intentionally allow it to be violated. Alternatively, intentional cooperation is required among a limited and readily identified set of people to break security. There is no inherent vested interest in this small group which would lead them to initiate or cooperate with a security break. If bribed, the possibility of discovery is significant and subsequent penalties would be severe.

Since the people who are capable of breaking the system are few, and since the cooperation of geographically dispersed and non-interested parties is required, we feel the automated accounting system is at least as secure as current, manual systems.

Getting into the Accounting System

In order to use the accounting system you must connect the terminal in your office to the ANTS system on the second floor of the Center for Advanced Computation. This is done by making a simple phone call. The next connection that must be made is an ARPA Network connection so that you can talk to Multics. A simple command to the ANTS system will instruct it to do this for you. Once you are connected to Multics you will have to log into Multics, identify yourself and give your password. Once you've logged into the Multics system, all that remains to do is to type the command to Multics that will start up the accounting system. When you are through with the accounting system you type the command "quit" to leave the accounting system, "logout" to leave Multics, and then hang up your phone. This sounds like a great deal of work to do and a lot to remember. Actually it is straight forward. It takes a small amount of time and less typing than some of the longer accounting system commands.

You do not have to be concerned about your computer terminal being compatible with Multics. The only system that it is important to be compatible with is ANTS. Currently, ANTS supports 10, 15, and 30 characters per second full duplex ASCII terminals. Examples of these kinds of terminals are all models of Teletypes and most of the thermal-printers (Texas Instruments, NCR, Teleterm, etc.). In order for your terminal to be able to connect to ANTS, it must be in full duplex mode. If you have a separate acoustical coupler (that little box that you put the phone in) it may have a separate switch which also has to be set to full duplex (as opposed to half duplex). Once you have done this the system will work with ANTS. If you are lucky enough to have a multi-speed terminal, set the speed to its fastest level (for most terminals this is 30 characters per second or 300 baud). Don't worry about the speed setting, ANTS handles a wide variety of speeds automatically. The faster speeds are simply for your own convenience.

Now that you have your terminal set at full duplex (and possibly the acoustic coupler too) and you have set your terminal speed, it is time to make a phone call. Using a normal desk telephone, dial 333-7086. The phone will ring once or twice then be automatically answered. You will hear a high pitched tone when ANTS answers the phone. At this time place the telephone hand set into the acoustic coupler (that's the little phone cradle on your terminal or in the separate acoustic coupler box). Make sure the phone is right side up. The cord should come out through a little notch which may even be labelled "cord". Somewhere on the terminal or the separate acoustic coupler there will be a little green light which may have the word "carrier" written under it. When the little light comes on, you are connected to ANTS.

The next thing to do is to help ANTS figure out what speed your terminal is. Start typing the upper case character "Q". Keep typing these Q's at the rate of about 1 per second until ANTS either sends you a nice long message telling you that ANTS is on the other end of the phone line or until it starts typing the Q's back to you. If it answers by typing Q's then hit a carriage return. It will tell you that the message full of Q's has "no destination" and has been discarded. It will then give you the ANTS hail. The hail tells you what version of the ANTS system is running and other information of absolutely no interest to you.

ANTS requires you to log in before you can talk to Multics. To do this type "↑LOGIN" followed by your name. There is no charge for using ANTS; your name is used only to identify users of ANTS.

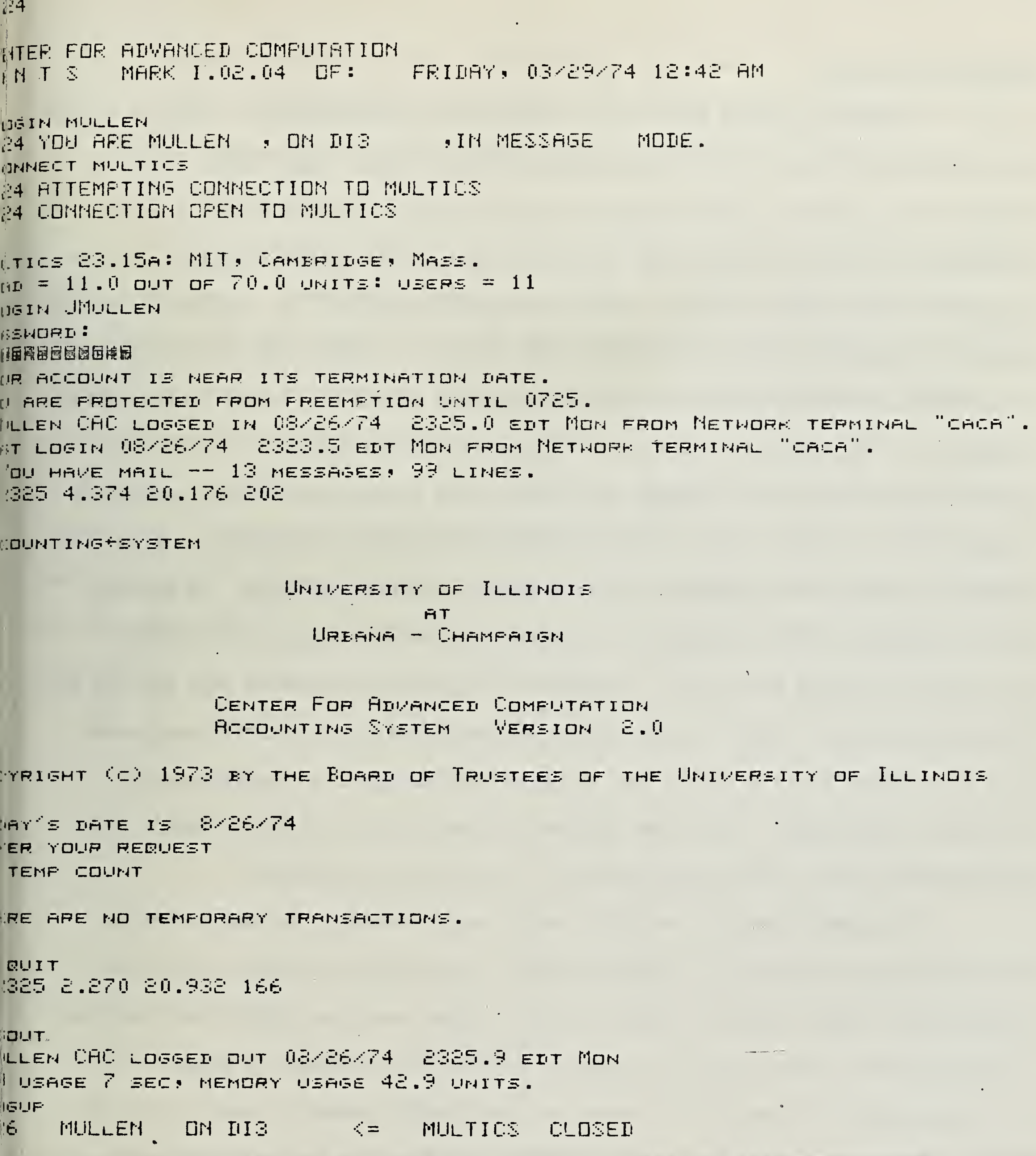
Now it is time to connect to Multics. That's simple.. Type "↑CONNECT MULTICS" That is a command to ANTS to connect you to Multics at MIT. ANTS will tell you when the connection is open to Multics and then Multics will respond by typing the Multics hail. The Multics hail is two lines long and will tell you what time of day it is, how many users (load units) are on the system, and which version of Multics is running. If nothing happened, you probably forgot to type the carriage return at the end of the ↑CONNECT MULTICS command - type the carriage return now.

After you receive the Multics hail, you are ready to login. To login, type the Multics command "login <registered user name>". Don't forget the carriage return! For example, a user whose registered name was "Corbally" would log into Multics by saying "login Corbally". Multics will ask you for your password, but, before you type it, it will completely black out the area into which you are going to type your password. Therefore, no one can ever see it. It is a good idea to change your password every week or two. Use a password that is easy to remember but hard for someone to guess. In particular, don't use passwords like your name, your wife's name, your dog's name, your kid's name, your department name, etc. Be original. Passwords can be changed at login time. If you want to change your password type the four characters "-cpw"). after your name on the login line (e.g. "login Corbally -cpw"). This tells Multics that you want to change your password. Multics will then blackout an area into which you will type your old password. It will then ask you for the new password and blackout another area into which you can type the new password. If it was successful, it will tell you that the password has been changed.

Now that you are logged into Multics it is time to start up the accounting system. To start up the accounting system simply type the Multics command "accounting_system". When you are through with the accounting system you will type the accounting system command "quit". This will bring you back to the point at which you were when you logged in. You must now type the command "logout" and Multics will automatically log you out and close your ARPA network connection. Once ANTS tells you that the connection to Multics has been closed, you can hang up your telephone. If the only Multics program you ever use is the accounting system, contact the CAC programmers. They will show you how to automatically start the accounting system right after login and to have Multics automatically log you out when you type the accounting system command "quit".

Time to review all this. The steps you have to go through are:

1. Set your terminal (and separate acoustical coupler if you have one) to full duplex
2. Set the terminal speed to that which you want to use
3. Dial 333-7086 to get ANTS
4. When you hear the high pitched tone put the telephone hand set into the acoustic coupler phone cradle (with the cord in the right direction)
5. Wait for the green carrier light to come on (either on your terminal or separate acoustic coupler)
6. Type upper case Q's until ANTS responds
7. Type ↑LOGIN <your name>
8. Type ↑CONNECT MULTICS (don't forget the carriage return)
9. Log into Multics
10. Initiate the accounting system (unless the accounting system initiation has been made automatic)
11. Do your accounting system work
12. Quit the accounting system
13. Log out of Multics (unless logout is automatic)
14. Wait for ANTS to tell you your Multics connection is closed
15. Hang up your telephone
16. Turn off the terminal (and the acoustic coupler if it is separate)



```
24

NTER FOR ADVANCED COMPUTATION
N T S   MARK I.02.04  OF:    FRIDAY, 03/29/74 12:42 AM

OGIN MULLEN
24 YOU ARE MULLEN   , ON DI3      ,IN MESSAGE   MODE.
ONNECT MULTICS
24 ATTEMPTING CONNECTION TO MULTICS
24 CONNECTION OPEN TO MULTICS

ltics 23.15a: MIT, Cambridge, Mass.
ad = 11.0 out of 70.0 units: users = 11
ogin JMullen
ssword:
[redacted]
ur account is near its termination date.
u are protected from preemption until 0725.
llen CAC logged in 08/26/74  2325.0 edt Mon from Network terminal "caca".
st login 08/26/74  2323.5 edt Mon from Network terminal "caca".
'ou have mail -- 13 messages, 99 lines.
325 4.374 20.176 202

counting+system

                    University of Illinois
                              at
                      Urbana - Champaign


               Center For Advanced Computation
               Accounting System    Version  2.0

yright (c) 1973 by the Board of Trustees of the University of Illinois

ay's date is  8/26/74
er your request
 temp count

re are no temporary transactions.

 quit
325 2.270 20.932 166

out
llen CAC logged out 08/26/74  2325.9 edt Mon
 usage 7 sec, memory usage 42.9 units.
gup
6   MULLEN   ON DI3       <=   MULTICS  CLOSED
```

Figure 6 shows a sample session using the accounting system. It covers steps 6 thru 13 above.

Where to Get Help

Computing systems never work the way they are supposed to. When they stop working correctly they do it in surprisingly devious ways. Most failures (called crashes) will occur when ANTS crashes or when the entire Multics system crashes. Hopefully, you will not experience too many errors in the actual accounting system program. When Multics crashes it will take one of two forms. 1) Multics crashed before you tried to connect to it and is still down. In this case, ANTS will tell you that Multics is dead when you try to connect to it and will abort the connection attempt. 2) Multics dies when you are inside the accounting system. ANTS will tell you by printing out a message that it has just closed your Multics connection (usually in the middle of one of your nice long printouts). When Multics dies it normally takes 15 to 20 minutes to get it back up and running again. So try connecting again in 15 or 20 minutes. If you are the nervous type, call 333-0707. Tell whoever answers who you are. Ask him if he knows when Multics will come up or if he can find out. Even if he is not an accounting system or Multics programmer he will probably be able to help you (usually by finding an accounting system or Multics programmer). If no one answers at 333-0707 try 333-8150. That's the ANTS machine room. Ask the ANTS operator. He may know the answer.

When ANTS dies your terminal simply stops working. You may be typing away furiously and nothing is being printed. A sure sign of failure is if your green carrier light goes out (this is usually caused when you jiggle the telephone in the acoustic coupler cradle or bang too hard on your terminal in anger). If you accidentally interrupted the phone call or if ANTS crashed, recovery is very rapid. Simply start back at dialing 333-7086. By the time you have dialed the number, ANTS has probably automatically recovered from any crash and is working again. If ANTS crashes on you, the ARPA Network will tell Multics about it. Multics will automatically quit the accounting system and log you out. To restart yourself, connect to Multics, login, and initiate the accounting system. You may have to re-enter the last command you typed. If you were entering transactions into the temporary buffer find the last transaction in the buffer and print it so you know where you left off.

For more serious problems or just confusing situations call the accounting system programmers. Their phone number is included with the primer. You should feel free to call this number. Don't be embarrassed because you think your question is stupid or because you feel you might be imposing upon the programmers. Unless we know those points of the system which confused you, we don't know how to fix it so the next user never has to ask that question. The programmers will be hard to impose upon. They are proud of their system and like to talk about it. They are much more likely to impose upon you. We have found that they tend to want to tell you far more than you ever cared to know about what you thought was a simple question.