INTERACTIVE CONSULTING
via Natural Language

Stuart C. Shapiro
Stanley C. Kwasny
Computer Science Department
Indiana University
Bloomington, Indiana

TECHNICAL REPORT No. 12
INTERACTIVE CONSULTING
VIA NATURAL LANGUAGE

STUART C. SHAPIRO
STANLEY C. KWASN

JUNE, 1974
INTERACTIVE CONSULTING
via Natural Language

Stuart C. Shapiro
Stanley C. Kwasny
Computer Science Department
Indiana University
Bloomington, Indiana

Abstract

Interactive Programming systems often contain help commands to give the programmer on-line instruction regarding the use of the various systems commands. We argue that it would be relatively easy to make these help commands significantly more helpful by having them accept requests in natural language. As a demonstration, we have provided Weizenbaum's ELIZA program with a script that turns it into a natural language system consultant.

Key Words and Phrases: interactive programming, time-sharing systems, natural language processing, computer assisted instruction.

Introduction

Many interactive systems include a mechanism for automatic dissemination of information regarding the use of its commands. Typically, the user gets this information by entering a basic "help" command and providing the name of the command he wants information about. For example, on the DECsystem-10 [3], the user may type HELP, and get information on the HELP commands; HELP*, and get the names of documented features; or HELP<name>, and get information on the feature <name>. Figure 1 shows the results of typing HELP and HELP* on the system available at Indiana University.

The problem with such help commands is that the user must know which command he wants information about. If, instead, he only knows what he wants to do and wants to find out the proper command to use, he is reduced to a sequence of guessing command names. Help commands should be more user-oriented, allowing the user to describe in his own terms what he wants to do. The system would interpret the request and provide information on how to accomplish the desired task.

Interactive systems consultants (help commands) are excellent applications for natural language understanding programs. Since the context which the systems consultant must deal with is limited, even unsophisticated natural language programs are capable of dealing with it. The ease with which such consultants may be programmed and their usefulness argue that large interactive systems be provided with natural language consultants.

A Natural Language Consultant

Lest the reader fear that we are proposing an extensive research project rather than a program well within the state of the art, let
HELP

HELP COMMAND (12/27/71) ====
THE HELP COMMAND PRINTS OUT HELPFUL DOCUMENTATION ON
VARIOUS SYSTEM FEATURES. THE COMMAND

HELP
WILL PRINT OUT THIS MESSAGE.

HELP +
WILL PRINT OUT THE NAMES OF ALL CURRENTLY AVAILABLE INFO.

HELP <NAME>
WILL LOOK FOR, AND PRINT OUT THE INFO ABOUT THE SYSTEM
FEATURE NAMED IN <NAME>, FOR EXAMPLE

HELP DIRECT
WILL PRINT OUT INFO ON THE DIRECTORY COMMAND.

ONLY THE FIRST 6 CHARACTERS OF THE ARGUMENT ARE
LOOKED AT; THEY MUST BE A-Z, 0-9, OR *.

HELP +

HELP IS AVAILABLE FOR THE FOLLOWING:

ABACUS BASIC BATCOM BLIS10 BOOT11 CDRSTK COBDDT COBOL
COBGR CREF DELFIL DIRECT DSKRAT DUMPER FAILSA FGEN
FILCOM FORTRA FUDGE2 GLOB GRIPR HELP IMPORT ISAM
LIBRARY LINK LPTSPL OMPUT OIPSER PIP PLTSPL PTPSPL
QUEUE QUELST REUN SETSRG SORT SOUP SPACE SPRINT
SYSDPY SYSERR TECO UMOU 274

THE MONITOR HAS THE FOLLOWING COMMANDS:

ASSIGN ATTACH BACKSP CCINTI CLOSE COMPIL CONTIN COPY
CORE CPUNCH CREATE CREF CSTART CTEST D DAYTIM
DOORE DOT DEASSI DEBUG DELETE DETACH DIRECT DISMOU
DSK DUMP E EDIT EOF EXECUT FILE FINISH
FUDGE GET HALT HELP INITIA ICONTI KJOB LABEL
LIST LOAD LOGIN MAKE MOUNT PJOB PLEASE PLOT
PRERER PRINT PROTEC PUNCH QUEUE R REASSI REENTE
RENAME RESCUR RESWIN RUN SAVE SCHEDU SEND SET
SKIP SSAVE START SUBMIT SYSTAT TECO TIME TPUNCH
TTY TYPE UNLOAD VERSID VI YDADS ZERO

THE MONITOR HAS THE FOLLOWING SET COMMANDS:

BLOCKS CPR CORMAX CORMIN CPU CTEST DATE DAYTIM
DENSIT DSKFUL DSKPRI HPO NOMESS OPR SCHEDU SPOOL
TIME TTY WATCH

Figure 1. Help on the DECsystem-10.
us explain the minimal requirements of a natural language understanding system and why the systems consultant is a good application.

We will say that a system understands natural language if a user who knows what the system is capable of but who had not been specifically trained in the system's input language (i.e. its domain of competence) can phrase an input to the system and, possibly after some clarificatory dialogue (see, for example, [1]), have his input satisfactorily handled. The sophistication and complexity required of the system depends on its domain of competence. Relatively sophisticated systems have been written to obey commands to manipulate blocks on a tabletop [12] and to retrieve scientific information on lunar rocks [13]. Newell, et al. [8] discuss varying degrees of sophistication needed for understanding spoken language for various tasks among which is the systems consultant. Their version of the systems consultant, called Voice-CC, requires a much more sophisticated system than ours because understanding spoken language is a more difficult and less understood task than understanding language written in machine-readable form. In one respect their task is easier because Voice-CC is to communicate with the user over a voice channel at the same time the user is trying to use the system over a conventional terminal. The system can know what the user has been doing and this can be a great help in understanding what he is asking. We are proposing a consultant which operates via standard terminals. We will discuss a consultant that is independent of the system monitor, so that it has no auxiliary source of information on what the user might be attempting. (Though if the consultant were part of the monitor, it could have this information.) On the other hand, since the user is using
the terminal to ask his questions, he is presumed to know such things as what the end of transmission character is, while the sample protocol in Newell, et al. [op. cit., pp. 69-71] has a significant number of interactions on such topics. In either case the task is much easier than a general natural language understanding system because the system's domain of competence is so limited, viz. the commands and features of the interactive system. We can assume that the user of the consultant wants information about these commands and that the request will be phrased in terms of the operations which they can perform. It is only necessary to recognize these terms and respond with a discussion of the relevant command and, possibly, related commands. The system need not understand the fine details of the user's request, just the gist of what he would like to do. Therefore, building the consultant is not much more difficult than writing a manual and providing a good index/thesaurus.

There is a controversy over whether natural language is an appropriate query language [5; 6; 7]. The opposing views seem to stress the ambiguities and general sloppiness of natural language. We trust that we have adequately explained that this is not an issue for the limited context we are discussing. There is another opposing view, however, that questions the usefulness of natural language input. This view is that habitual users of any system will prefer to use a terse formal language rather than natural language, which is generally verbose. The common response to this is that natural language input is best suited to "casual users". But do casual users exist? If so, who are they and what systems do (would) they use? Our answer is, "We have met the casual users and they are we." Experienced programmers,
when faced with a new system or with the need to use an unfamiliar feature on their old system are casual users of the "help" program (system consultant). They use the system consultant because they do not know the command language, and they use it only until they learn the command language. What such a user wants is to be able to describe the operation he would like to perform and to be told the correct command to use in the given system. This is the natural language system consultant we are proposing.

The ELIZA Helper

A natural language system consultant has been described briefly elsewhere [9]. To further demonstrate its feasibility, we have made Weizenbaum's ELIZA program [4; 10; 11] into a partial implementation.

The ELIZA program is actually quite simple and involves no sophisticated parsing, analysis or "understanding" of language. The input sentence is searched for predefined keywords, substitutions are made where specified, and the sentence is broken into phrases which can be used in the output sentence. Associated with each keyword is a level number which determines the preference of a response related to that keyword along with a list of reassembly rules to be used in the response. As the input sentence is scanned, a list of keywords in the sentence is constructed with the most preferred keyword at the front of the list. When the scan is completed, the decomposition pattern for the most preferred keyword is applied to the input sentence. If this pattern matches, parts of the input may be concatenated with a rule for that keyword to form the output sentence. Where necessary, subsequent keywords from the input sentence are used. If no keywords
are found or all decomposition patterns fail to match, a stock sentence is chosen as the response.

The keywords, preference numbers, decomposition patterns and reassembly rules are written on a file called the "script" which is read and used by the ELIZA program itself. By changing scripts, ELIZA can be made to perform different roles. ELIZA has been made to perform the system consultant role by choosing the keywords from the commands, uses of the commands and their synonyms. The reassembly rules are paraphrases or quotes from sections of the manual designed to provide information about the system and to answer the presumed questions.

The current implementation uses a SNØBØL4 version of ELIZA [4] (we actually use Cal SNØBØL, a smaller, faster version of SNØBØL4) as a consultant for the KRONOS 2.1 operating system on the CDC 6600 computer at Indiana University. Specifically, the script is designed to provide help with permanent file (PFILE) commands used interactively, as well as knowledge about logging off the system, submitting files of control cards to the input queue, routing files to printers or card punches at various remote facilities, recovering an interactive job after being disconnected from the system, copying files, and other useful procedures as described in CDC's Time-Sharing User's Manual [2].

A user accesses the consultant by typing -HELP which activates a KRONOS procedure file, which then calls the ELIZA helper. It also copies every conversation for later study. Such sample conversations permit the maintenance of the script in a dynamic state to fit the specific needs of users and to provide navigational tools in the task of predicting conversation flow. A typical conversation is shown in Figure 2.


-A HELP
A COPY OF THIS CONVERSATION IS BEING
MADE FOR FURTHER STUDY...

TO TERMINATE THIS PROGRAM, TYPE QUIT,
THANKS, OR THANK YOU...

HOW DO YOU DO. WHAT IS YOUR PROBLEM?

I NEED SOME INFORMATION ABOUT PFIL~ ES.
FILES MAY BE ASSOCIATED WITH A JOB BY USING EITHER
THE ATTACH (DIRECT FILE) OR GET (INDIRECT FILE)
COMMANDS. FILES MAY BE MADE PERMANENT BY USING EITHER
THE DEFINE (DIRECT FILE) OR SAVE (INDIRECT FILE) COMMANDS

I'M NOT SURE IF I WANT DIRECT OR INDIRECT FILES.
A DIRECT ACCESS FILE IS A PERMANENT FILE WHICH, UPON
REQUEST, BECOMES ASSOCIATED DIRECTLY WITH THE USERS
JOB. UNLIKE REQUESTS FOR INDIRECT ACCESS FILES, A
COPY OF THE PERMANENT FILES IS NOT CREATED FOR USER
ACCESS. USE OF A DIRECT ACCESS FILE IS REQUESTED BY
ENTERING THE ATTACH COMMAND, AND ALL SUBSEQUENT OPERATIONS
ARE PERFORMED ON THE PERMANENT FILE ITSELF. A DIRECT
ACCESS FILE IS CREATED BY USE OF THE DEFINE COMMAND

How is that different from an indirect access file?
AN INDIRECT ACCESS FILE IS A PERMANENT FILE THAT CANNOT
BE ACCESSED DIRECTLY BY THE USER. WHEN ACCESS IS REQUESTED
BY A GET COMMAND, A COPY OF THE PERMANENT FILE IS
CREATED AND ASSOCIATED WITH THE JOB MAKING THE REQUEST.
NOTE THAT IF THE WORKING FILE IS ALTERED, THE USER
MUST ENTER THE REPLACE COMMAND TO UPDATE THE PERMANENT
FILE. AN INDIRECT ACCESS FILE IS CREATED BY USE OF
THE SAVE COMMAND

WHAT ELSE CAN YOU TELL ME ABOUT INDIRECT ACCESS FILES?
CAN YOU BE MORE SPECIFIC?

Suppose I want to save indirect files,
THE SAVE COMMAND CREATES AN INDIRECT ACCESS PERMANENT
FILE. THIS COMMAND ALLOWS A USER TO RETAIN A COPY
OF THE SPECIFIED WORKING FILE IN THE PERMANENT FILE
SYSTEM

How do I use the save command?
THE FORM OF THE SAVE COMMAND IS -
SAVE, LFN=PFN, PW=PASSWORD, CT=N, M=M

WHAT IS MEANT BY CT, I DON'T UNDERSTAND.
CT REPRESENTS THE PERMANENT FILE CATEGORY. THE FILE
CATEGORY DETERMINES WHICH USERS MAY ACCESS A FILE.
THERE ARE THREE CATEGORIES OF PERMANENT FILES - PRIVATE,
SPRIV, AND PUBLIC

Figure 2a. A session with the ELIZA Helper. User's input is on
lines beginning with "/" or "/".
WHO MAY USE A FILE IF IT IS SAVED AS A PUBLIC FILE?
PU REPRESENTS A PUBLIC FILE. THESE FILES ARE AVAILABLE
FOR ACCESS BY ALL USERS KNOWING THE FILE NAME, PASSWORD,
AND USER NUMBER. ONLY THE TOTAL NUMBER OF ACCESSES
TO FILES OF THIS CATEGORY ARE RECORDED FOR THE ORIGINATOR
OF THE FILE.

PLEASE TELL ME WHAT YOU MEAN BY M=M ON THE SAVE COMMAND.
M REPRESENTS THE FILE OR USER PERMISSION MODE. THERE
ARE SEVERAL POSSIBLE MODES - WRITE, MODIFY, APPEND,
READ, READMD, READAP, EXECUTE, OR NULL.

WHAT AM I ABLE TO DO WITH A FILE SAVED IN WRITE MODE?
W REPRESENTS THE WRITE MODE AND ALLOWS THE USER TO WRITE, MODIFY, APPEND, READ, EXECUTE, OR PURGE THE FILE.

YOU MENTIONED WORKING FILES EARLIER AND I DIDN'T UNDERSTAND.
A WORKING FILE IS EITHER A NEW FILE, CREATED BY THE USER, OR A COPY OF AN INDIRECT ACCESS FILE THAT ALREADY EXISTS IN THE SYSTEM. ALL WORKING FILES ARE TEMPORARY IN NATURE AND CAN EXIST NO LONGER THAN THE USER IS LOGGED INTO THE SYSTEM. A WORKING FILE IS A LOCAL FILE. THE PARAMETER LFN ON MOST TIME-SHARING JOB COMMANDS AND PERMANENT FILE COMMANDS SIGNIFIES A LOCAL FILE NAME.

YESTERDAY I LOST A FILE BECAUSE I HAVN'T USED IT IN AWHILE.
DO YOU WANT TO KNOW ABOUT PRGLIST, WHICH GIVES A LISTING OF YOUR PURGED FILES, OR DO YOU WANT TO KNOW ABOUT PRGCOPY, WHICH WILL ALLOW YOU TO RETRIEVE A COPY OF A PURGED FILE?

TELL ME ABOUT PRGCOPY.
PRGCOPY IS A ROUTINE WHICH ALLOWS THE USER TO RETRIEVE A PERMANENT FILE WHICH HAS BEEN PURGED IN ACCORDANCE WITH THE PFILE PURGING ALGORITHM.

HOW DO I USE PRGCOPY?
THE USER MAY ACCESS THE PRGCOPY ROUTINE BY THE COMMAND PRGCOPY, DATE, PFNAME. WHERE DATE REPRESENTS THE NUMERIC DAY OF THE MONTH WHEN THE FILE WAS PURGED, AND PFNAME REPRESENTS THE NAME OF THE PF FILE WHICH IS TO BE RECOVERED. THE DATE IS A REQUIRED PARAMETER AND IS AVAILABLE FROM PRGLIST. IF PFNAME IS OMITTED, ALL OF THE USERS PF FILES ARE RECOVERED FROM THAT DAYS PURGE TAPE. NOTE THAT PF FILES MAY NOT BE RECOVERED AFTER ONE MONTH. ALSO NOTE THAT TO ACCOMPLISH THIS FEAT FROM THE TELETYPewriter, A SUBMIT COMMAND MUST BE USED.

Figure 2b. A session with the ELIZA Helper.
? What does the Submit command do?

The Submit command allows a validated user to create a batch job deck image and submit it to the local batch queue as a deferred batch job. The user cannot interact with a job submitted for processing using this command. Rather, the job is treated in the same manner as a job submitted from the card reader at WCC, Balantine, Memorial, or Business.

? How do I submit a job in this way?

The form of the Submit command is:

```
SUBMIT(LFN, Q, NR)
```

? What is Q?

Q represents the queue specified in a submit command. It may be either 'B', 'N', or 'E'. B means that the job output is disposed to the batch queue to be printed, punched, etc., at WCC. N means that the job output is disposed to the batch queue, but is dropped at job termination. (Default value). E means that the job output is disposed to export/import queue for printing at a remote site.

? What is NR?

NR refers to the no rewind option. If present, this parameter will inhibit rewind after execution of associated command.

? One more question - How do I turn this machine off?

To logoff from telex, type goodbye or bye. An alternate way to logoff is through the hello or login commands. These have the same effect as the bye command except the login sequence is reinitialized.

? Thank you for your help.

My pleasure.

EXIT.

---

Figure 2c. A session with the ELIZA Helper.
Organization of the consultant script follows the general theme that when a command name, a synonym of the command name or a word implying some use of that command is recognized in a user input, the user is presumed to be asking for information about that command. The initial response is a general description of the usage of the command. It gives enough information that the user's question is probably answered or at least the proper terminology is provided to rephrase the question.

If the same keyword reappears, the system responds with more specific information until the feature is completely described. The next use of the keyword is responded to with:

- CAN YOU BE MORE SPECIFIC? or
- PLEASE DESCRIBE YOUR PROBLEM IN MORE DETAIL or
- WHAT DO YOU MEAN BY _______? I DON'T UNDERSTAND

where ______ represents the input string. Further uses of the keyword are ignored, allowing less preferred keywords to determine the response.

Preference numbers determine dominance among keywords. Requests for information about parameters on control cards always dominate, since these keywords have a higher precedence than the simple name of a control card. If an input sentence were:

What does PW=PASSWORD mean on an ATTACH card?

the system would respond relative to the keyword PW and describe what password should be specified when manipulating a file rather than explaining more about the ATTACH command itself.
A more problematic situation occurs whenever the same keyword has differing interpretations, depending on the context. A partial solution is provided by assuming the user will remain within the overall context of a given script (an underlying assumption throughout ELIZA's history). Even with this assumption ambiguities arise. For example, the permanent file structure under KRONOS permits the specification of a mode under which a file may be accessed. These include a READ, WRITE, and APPEND mode. But in many situations, an input sentence may contain one of these keywords, though the user is not requesting mode information. An answer to this problem is provided in the ELIZA system through the use of more complex decomposition patterns. A phrase such as READ MODE may be specified as part of the pattern associated with the keyword READ so that responses relative to that word are not given indiscriminately. It is important to note that such disambiguation cannot always be accomplished in this manner. In some instances, ELIZA is made to respond with a question formulated to resolve the ambiguous keyword. For example, if an input sentence were:

How do I find the turnaround time at Marshal H. Wrubel Computing Center?

the system would respond:

WOULD YOU LIKE TO DROP OR SUBMIT A JOB OR WOULD YOU LIKE TO SEE A STATUS OF THE QUEUES AT WCC?

Thus, a user is encouraged to use unambiguous keywords and is led to the eventual solution to his problem.
Summary

An excellent application for natural language understanding systems is an interactive system consultant. This is true for several reasons. The user of a system consultant is, ipso facto, not well versed in the system command language, and will cease using the consultant precisely when he does learn the command language. He is, therefore, precisely the kind of user best served by a natural language input system. On the other hand, the system consultant operates on a very restricted domain, viz. the system commands and the uses to which they may be put. At this time, natural language understanding systems have been successful when applied to restricted domains and they have been successful only in such applications. Furthermore, the system consultant does not require a fine understanding of the input. It is acceptable if the consultant merely recognizes what command or feature is being inquired about and launches into a discussion of that feature. To demonstrate the feasibility of a natural language system consultant, we have implemented one using ELIZA, a keyword oriented conversation program.
References


6. Hill, I.D. Wouldn't it be nice if we could write computer programs in ordinary English—or would it? The Computer Bulletin 16, 6 (June, 1972), 306-12.


11. ----. Contextual understanding by computers. CACM 10, 8 (August, 1967), 474-80.


Appendix I: Keywords used for ELIZA Helper

<table>
<thead>
<tr>
<th>APPEND</th>
<th>EDITOR</th>
<th>MODE</th>
<th>RA</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT</td>
<td>EJ</td>
<td>MODIFY</td>
<td>READ</td>
</tr>
<tr>
<td>AT=AT</td>
<td>ELIZA</td>
<td>NAME</td>
<td>READAP</td>
</tr>
<tr>
<td>ATTACH</td>
<td>END</td>
<td>NAMES</td>
<td>READMD</td>
</tr>
<tr>
<td>BA</td>
<td>EQ</td>
<td>NEVER</td>
<td>RECORD</td>
</tr>
<tr>
<td>BAL</td>
<td>EQ=EQ</td>
<td>NEW</td>
<td>RECORDS</td>
</tr>
<tr>
<td>BALLANTINE</td>
<td>EVICT</td>
<td>NFN</td>
<td>RECOVER</td>
</tr>
<tr>
<td>BALLINTINE</td>
<td>EXECUTE</td>
<td>NFN=OFN</td>
<td>RECOVERY</td>
</tr>
<tr>
<td>BRF</td>
<td>F=LFN</td>
<td>NO</td>
<td>RELEASE</td>
</tr>
<tr>
<td>BUSINESS</td>
<td>FILE</td>
<td>NODROP</td>
<td>REPLACE</td>
</tr>
<tr>
<td>BYE</td>
<td>FILES</td>
<td>NR</td>
<td>RETRIEVE</td>
</tr>
<tr>
<td>C</td>
<td>FN</td>
<td>OPF</td>
<td>RETURN</td>
</tr>
<tr>
<td>C=CC</td>
<td>FN=PFN</td>
<td>OLD</td>
<td>RM</td>
</tr>
<tr>
<td>CAN'T</td>
<td>GET</td>
<td>OPERATING</td>
<td>ROUTE</td>
</tr>
<tr>
<td>CATALOG</td>
<td>GOODBYE</td>
<td>OPTIONS</td>
<td>S</td>
</tr>
<tr>
<td>CATALOGUE</td>
<td>HALT</td>
<td>P</td>
<td>SAVE</td>
</tr>
<tr>
<td>CATEGORY</td>
<td>HELP</td>
<td>PARAMETER</td>
<td>SEMI</td>
</tr>
<tr>
<td>CATLIST</td>
<td>I'M</td>
<td>PARAMETERS</td>
<td>SEMI-PRIVATE</td>
</tr>
<tr>
<td>CC</td>
<td>INDIRECT</td>
<td>PASSWORD</td>
<td>SEND</td>
</tr>
<tr>
<td>CDC</td>
<td>KRONOS</td>
<td>PASSWORDRD</td>
<td>SORRY</td>
</tr>
<tr>
<td>CERTAINLY</td>
<td>L=LOC</td>
<td>PERMIT</td>
<td>SPEC</td>
</tr>
<tr>
<td>CHANGE</td>
<td>LFN</td>
<td>PRG</td>
<td>SPRIV</td>
</tr>
<tr>
<td>COMPUTER</td>
<td>LFN=PFN</td>
<td>PERMANENT</td>
<td>STATUS</td>
</tr>
<tr>
<td>COMPUTERS</td>
<td>LIST</td>
<td>PERMIT</td>
<td>STOP</td>
</tr>
<tr>
<td>COPY</td>
<td>LISTING</td>
<td>PFILE</td>
<td>SUBMIT</td>
</tr>
<tr>
<td>COPYBF</td>
<td>LNH</td>
<td>PFILES</td>
<td>SYSTEM</td>
</tr>
<tr>
<td>COPYBR</td>
<td>LO</td>
<td>PW</td>
<td>TELEX</td>
</tr>
<tr>
<td>COPYSBF</td>
<td>LO=OPTIONS</td>
<td>WRUBEL</td>
<td>UN</td>
</tr>
<tr>
<td>COPYCF</td>
<td>LOC</td>
<td>WRUBLE</td>
<td>UN=USERNUM</td>
</tr>
<tr>
<td>COPYCR</td>
<td>LOCATION</td>
<td>XXX</td>
<td>W</td>
</tr>
<tr>
<td>CT</td>
<td>LOCATIONS</td>
<td>WCC</td>
<td>WAT</td>
</tr>
<tr>
<td>CT=N</td>
<td>LOGOFF</td>
<td>WON'T</td>
<td>WORKING</td>
</tr>
<tr>
<td>DEFINE</td>
<td>LOST</td>
<td>WRITE</td>
<td>WRITE</td>
</tr>
<tr>
<td>DIRECT</td>
<td>M</td>
<td>WRUBL</td>
<td>WRUBL</td>
</tr>
<tr>
<td>DISPOSE</td>
<td>M=M</td>
<td>XXX</td>
<td>WRUBL</td>
</tr>
<tr>
<td>DON'T</td>
<td>MACHINE</td>
<td>YES</td>
<td>WRUBL</td>
</tr>
<tr>
<td>DROP</td>
<td>MACHINES</td>
<td>YOU</td>
<td>WRUBL</td>
</tr>
<tr>
<td>DROPPED</td>
<td>MAYBE</td>
<td>6600</td>
<td>WRUBL</td>
</tr>
<tr>
<td>E</td>
<td>ME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EDIT</td>
<td>MEM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MEMORIAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MRF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix II: Script for ELIZA Helper

AM     L/2/ S/AIR/
APPEND L/21/ D/APPEND MODE /
'A' REPRESENTS THE APPEND MODE WHICH ALLOWS THE USER TO APPEND
INFORMATION AT THE END OF THE FILE (EOI)':'CF SPEC:NEWKEY:NEWKEY:
AT=AT L/21/ D/CF AT: 
AT L/1/ D //
'AT' REFER TO THE ATTRIBUTE OF THE EQUIPMENT TO BE USED WHEN ROUTING
A FILE. THE DEFAULT IS NONE. CURRENTLY, THE ONLY USE OF THIS FEATURE
IS TO SPECIFY THE UCS PRINT CHAIN AT BALLANTINE':'NEWKEY:NEWKEY:NEWKEY:
ATTACH L/20/ D //
CF FILE:'THE FORM OF AN ATTACH COMMAND IS -
ATTACH,LFN=PFN/UN=USERNUM,PW=PASSWORD':'CF SPEC:NEWKEY: 
BA L/19/ D //CF BALLANTINE: 
BAL L/19/ D //CF BALLANTINE: 
BRF L/19/ D //CF BALLANTINE: 
BALLANTINE L/19/ D //CF BALLANTINE: 
BALLANTINE L/19/ D //CF BALLANTINE: 
BALLANTINE L/19/ D //
'DO YOU WANT TO SUBMIT, ROUTE, OR DROP A JOB AT BALLANTINE, OR DO YOU
WANT TO SEE A STATUS OF JOBS IN THE BALLANTINE QUEUES?':'CF SPEC:NEWKEY: 
BUSINESS L/19/ D //
'DO YOU WANT TO SUBMIT, ROUTE, OR DROP A JOB AT BUSINESS, OR DO YOU
WANT TO SEE A STATUS OF JOBS IN THE BUSINESS QUEUES?':'CF SPEC:NEWKEY: 
BYE L/20/ D //CF LOGOFF: 
C L/21/ D //
'C' REFER TO THE ESCAPE CHARACTER USED TO IDENTIFY REFORMATTING
DIRECTIVES IN THE FILE TO BE SUBMITTED UNDER A SUBMIT COMMAND. IF
OMITTED, THE SYSTEM ASSUMES C=':CF NEWKEY: 
CATALOG L/19/ D //CF CATLIST: 
CATALOGUE L/19/ D //CF CATLIST: 
CATEGORY L/20/ D //CF CT: 
CATLIST L/20/ D //
'THE CATLIST COMMAND SELECTS A LISTING OF PERTINENT INFORMATION ABOUT
EACH FILE IN THE USERS CATALOG. IF AN ALTERNATE USER NUMBER IS SPECIFIED,
THE USER OBTAINS A LISTING OF ALL FILES THAT HE CAN ACCESS IN THE
ALTERNATE USERS CATALOG':'THE FORM OF THE CATLIST COMMAND IS -
CATLIST/LO=OPTIONS,FN=PFN,UN=USERNUM,L=LISTING:CF SPEC:NEWKEY: 
C=CC L/21/ D //CF CC: 
CDC L/18/ D //CF 6600: 
CC L/21/ D //
'CC' REFER TO THE COPY COUNT ON A ROUTE CARD. THIS MUST BE EXPRESSED
AS A DECIMAL NUMBER BETWEEN 1 AND 63 INCLUSIVE. THE DEFAULT VALUE IS 1':
CERTAINLY L/0/ D //CF YES: 
CHANGE L/20/ D //
'THE CHANGE COMMAND ALLOWS THE ORIGINATOR OF A FILE TO ALTER ANY OF
SEVERAL PARAMETERS WITHOUT HAVING TO ATTACH AND REDEFINE THE FILE OR
RETRIEVE AND SAVE IT':'THE FORM OF THE CHANGE COMMAND IS -
CHANGE,NFN=OFN,PW=PASSWORD,CT=N,M=M:CF SPEC:NEWKEY:
WHAT DO YOU THINK MACHINES HAVE TO DO WITH YOUR PROBLEM?:

DO YOU THINK COMPUTERS CAN HELP PEOPLE?:

DO COMPUTERS WORRY YOU?: WHY DO YOU MENTION COMPUTERS?:

WHAT IS IT ABOUT MACHINES THAT WORRIES YOU?:

WHAT DO YOU THINK ABOUT MACHINES?:

COMPUTERS

COPY

COPYING ONE FILE TO ANOTHER MAY BE ACCOMPLISHED BY USE OF ANY OF THE FOLLOWING COMMANDS - COPYBF, COPYBR, COPYSBF, COPYCF, COPYCR, OR COPYSCF:

COPYBF, COPYBR, AND COPYSBF ALLOW THE USER TO DUPLICATE A FILE OR RECORD. COPYBF CAN BE USED TO COPY MOST FILES IF AN EXACT COPY IS DESIRED. THE DIFFERENCE BETWEEN COPYC AND COPYB IS THE PARITY ON TAPE COPIES. COPYB IS USED TO COPY UP TO THE FIRST RECORD MARK ENCOUNTERED ON THE FILE, AND COPYSBF IS THE SAME AS COPYBF, BUT THE COPY IS SHIFTED RIGHT BY ONE CHARACTER, THUS AVOIDING THE CARRIAGE CONTROL CHARACTER:

THE FORM OF COPY COMMANDS IS - COPY**,NAME1,NAME2. WHERE COPY** REFERS TO THE APPROPRIATE COPY COMMAND:

COPYBF

COPYCF

COPYCR

COPYCSF

COPYBF

COPYCF

COPYCR

COPYCSF

CT

CT

CT REPRESENTS THE PERMANENT FILE CATEGORY. THE FILE CATEGORY DETERMINES WHICH USERS MAY ACCESS A FILE. THERE ARE THREE CATEGORIES OF PERMANENT FILES - PRIVATE, SPRIV, AND PUBLIC:

DIRECT

A DIRECT ACCESS FILE IS A PERMANENT FILE WHICH, UPON REQUEST, BECOMES ASSOCIATED DIRECTLY WITH THE USERS JOB. UNLIKE REQUESTS FOR INDIRECT ACCESS FILES, A COPY OF THE PERMANENT FILES IS NOT CREATED FOR USER ACCESS. USE OF A DIRECT ACCESS FILE IS REQUESTED BY ENTERING THE ATTACH COMMAND, AND ALL SUBSEQUENT OPERATIONS ARE PERFORMED ON THE PERMANENT FILE ITSELF. A DIRECT ACCESS FILE IS CREATED BY USE OF THE DEFINE COMMAND:
THE DEFINE COMMAND ALLOWS A USER TO CREATE A DIRECT ACCESS PERMANENT FILE (PFN) AND ATTACH IT IN WRITE MODE. THE FORM OF THE DEFINE COMMAND IS:

\[ \text{DEFINE, LFN=PFN/PW=PASSWRD, CT=N, M=M} \]

DROP

A JOB MAY BE DROPPED AT SEVERAL PLACES AROUND CAMPUS AND RETURNED BY WAY OF THE COURIER SERVICE. THIS IS ACCOMPLISHED BY SPECIFYING A DROP PARAMETER ON THE JOB CARD. THIS PARAMETER CONSISTS OF ONE OF THE FOLLOWING:

- DSW(SWAIN)
- DCH(Chemistry)
- DEI(Education)
- DMA(Memorial)
- DLI(Lindley)
- DLI(Ballantine)
- DL(Davidson)
- DLI(Lindley)

DROP CAN BE ANY OF THE FOLLOWING:

- V(SWAIN)
- C(Chemistry)
- X(Education)
- A(Memorial)
- E(Ballantine)
- Z(Lindley)

Dropped

EDIT

EDITOR

EJ

WEN EJ APPEARS ON A ROUTE CARD, THE FILE TO BE ROUTED WILL BE SENT ONLY AT THE COMPLETION OF THE JOB IN WHICH THE COMMAND OCCURS.

EJ MEANS END-OF-JOB.

ELIZA

HOW DID YOU KNOW MY NAME?

EQ=EQ

EQ

EQ REFERS TO THE EQUIPMENT SPECIFIED ON A ROUTE COMMAND. THE DEFAULT FOR THIS PARAMETER IS ANY PRINTER, BUT THE USER MAY SPECIFY OTHER DEVICES:

- SOME OF THE EQUIPMENT THAT MAY BE SPECIFIED ON A ROUTE CARD IS 501 FOR SPECIFYING THE CDC 501 PRINTER AT WCC, 1403 FOR THE IBM 1403 PRINTER AT WCC, PH FOR SPECIFYING THAT A FILE IS TO BE PUNCHED AS A HOLLERITH FILE, PT FOR SPECIFYING THAT THE FILE IS TO BE PUNCHED ON PAPER TAPE, PL FOR SPECIFYING THAT THIS FILE IS A PLOT.

E

EXECUTE

EVICT

EXECUTE

EXECUTE MODE

EXECUTE MODE AND ALLOWS THE USER TO EXECUTE THE FILE.

FILE

FILES MAY BE ASSOCIATED WITH A JOB BY USING EITHER THE ATTACH (DIRECT FILE) OR GET (INDIRECT FILE) COMMANDS. FILES MAY BE MADE PERMANENT BY USING EITHER THE DEFINE (DIRECT FILE) OR SAVE (INDIRECT FILE) COMMANDS.
GET FILE: THE FORM OF A GET COMMAND IS

GET,LFN=PFN/UN=USERNUM,PW=PASSWORD':CF SPEC:NEWKEY/

HALT L /19/ D //CF STOP/
GOODBYE L /20/ D //CF LOGOFF/
HELP L /17/ D //

'CAN YOU DESCRIBE YOUR PROBLEM?': 'HOW CAN I HELP YOU?'

'WHAT IS REALLY YOUR PROBLEM?': 'PLEASE STATE YOUR PROBLEM IN A DIFFERENT WAY'

'I CANT HELP YOU WITH THAT PROBLEM':

'WOULD YOU LIKE TO KNOW MORE ABOUT PFFILES?': CF SPEC://

I'M S /YOU ARE//
INDIRECT L /21/ D //

'AN INDIRECT ACCESS FILE IS A PERMANENT FILE THAT CANNOT BE ACCESSED
DIRECTLY BY THE USER. WHEN ACCESS IS REQUESTED BY A GET COMMAND, A
COPY OF THE PERMANENT FILE IS CREATED AND ASSOCIATED WITH THE JOB MAKING
THE REQUEST. NOTE THAT IF THE WORKING FILE IS ALTERED, THE USER MUST
ENTER THE REPLACE COMMAND TO UPDATE THE PERMANENT FILE. AN INDIRECT
ACCESS FILE IS CREATED BY USE OF THE SAVE COMMAND': CF SPEC:NEWKEY://
KRONOS L /17/ D //

'KRONOS IS THE OPERATING SYSTEM USED AT I.U. ON THE CDC 6600':

'KRONOS WAS A TITAN AND THE FATHER OF ZEUS': CF SPEC:NEWKEY://
F=LFN L /21/ D //CF LFN://
L=LOC L /21/ D //CF LOC://
LFN=PFN L /21/ D //CF LFN:CF PFN://
LFN L /21/ D // 'LFN REPRESENTS LOCAL FILE NAME (PRIMARY OR WORKING FILE)//
LIST L /21/ D //

'THE LIST COMMAND PRINTS THE CONTENTS OF THE PRIMARY FILE AT THE
TERMINAL. THE FORM OF THE LIST COMMAND IS - LIST,XXX,R,F=LFN. IF THE
F OPTION IS SPECIFIED, THEN WORKING FILE LFN IS PRINTED': CF SPEC:NEWKEY://
LISTING L /16/ D //CF CATLIST://
LNH L /20/ D //

'LNH IS AN OPTIONAL FORM OF THE LIST COMMAND AND HAS THE SAME PARAMETERS.
THE LISTING IS MADE WITHOUT A HEADER. THE FORM OF THE LNH COMMAND IS -
LNH,XXX,R,F=LFN': CF SPEC:NEWKEY://
LO=OPTIONS L /21/ D //CF LO://
LO L /21/ D //

'LO REFERS TO LIST OPTIONS ON THE CATLIST COMMAND. IT MAY BE SET TO
ANY OF THE FOLLOWING - F(FULL), FP(PERMISSION INFORMATION ONLY), P
LIST OF USERNUMBERS THAT HAVE ACCESSED THE FILE), OR 0(ZERO):/
LOC L /21/ D //

'LOC REPRESENTS A REMOTE LOCATION. THIS MAY BE EITHER
BAL(BALLANTINE), MEM(MEMORIAL), OR BUS(BUSINESS). IF SPECIFYING A
LOCATION ON A ROUTE COMMAND, ANY OF THE FOLLOWING MAY BE USED -
WCC(WRUBEL), IUPUI(IU/PU INPLS), BAL(BALLANTINE), CHEM(CHEMISTRY),
MEM(MEMORIAL), IUMW(GARY), IUSB(SOUTH BEND), BUS(BUSINESS), IU
SE(JEFFERSONVILLE), IUFW(FORT WAYNE), LILLY(ELI LILLY CO.)': CF SPEC:NEWKEY://
LOCATION L /19/ D //CF LOC://
LOCATIONS L /19/ D //CF LOC://
LOGOFF L /20/ D //

'TO LOGOFF FROM TELEX, TYPE GOODBYE OR EYE. AN ALTERNATE WAY TO LOGOFF IS THROUGH THE HELLO OR LOGIN COMMANDS. THESE HAVE THE SAME EFFECT AS THE EYE COMMAND EXCEPT THE LOGIN SEQUENCE IS REINITIALIZED':CF SPEC:NEWKEY://
LOST L /19/ D //CF PURGED://
M=M L /21/ D //CF MODE://
M L /21/ D //CF MODE://
MACHINE L /10/ D //CF COMPUTER://
MACHINES L /10/ D //CF COMPUTER://
MAYBE L /2/ D //CF PERHAPS://
ME $ /YOU://
MEM L /19/ D //CF MEMORIAL://
MRF L /19/ D //CF MEMORIAL://
MEMORIAL L /19/ D //

'DO YOU WANT TO SUBMIT, ROUTE, OR DROP A JOB AT MEMORIAL, OR DO YOU WANT TO SEE A STATUS OF JOBS IN THE MEMORIAL QUEUES?' :CF SPEC:NEWKEY://
MODE L /20/ D //
'M REPRESENTS THE FILE OR USER PERMISSION MODE. THERE ARE SEVERAL POSSIBLE MODES - WRITE, MODIFY, APPEND, READ, READMD, READAP, EXECUTE, OR NULL':CF SPEC:NEWKEY:NEWKEY://
MODIFY L /21/ D //MODIFY MODE://
'M REPRESENTS THE MODIFY MODE WHICH ALLOWS THE USER TO MODIFY INFORMATION WITHIN A DIRECT ACCESS FILE AND/OR APPEND INFORMATION AT THE END OF THE FILE. THE USER MAY ALSO READ OR EXECUTE THE FILE':CF SPEC:NEWKEY://
NAME L /15/ D //
'I AM NOT INTERESTED IN NAMES': 'I'VE TOLD YOU BEFORE, I AM NOT INTERESTED IN NAMES - PLEASE CONTINUE':CF HELP://
NAMES L /15/ D //CF NAME://
NEVER L /0/ D //
'NEVER?':NEWKEY:NEWKEY:NEWKEY:NEWKEY://
NEW L /20/ D //
'THE NEW COMMAND ALLOWS THE USER TO CREATE A NEW PRIMARY FILE.
THE FORM OF THIS COMMAND IS NEW,LFN. THE FILE NAME SPECIFIED BECOMES THE NEW PRIMARY FILE AND ALL CURRENT WORKING FILES ARE RELEASED UNLESS
NO =OFN IS THE NEXT COMMAND ENTERED':CF SPEC:NEWKEY://
NFM=OFN L /21/ D //CF NFM://
NFM L /21/ D //NFM REPRESENTS NEW FILE NAME IN CHANGE COMMAND://
NO L /0/ D //
'WHY -NO- ?':NEWKEY:'WHY NOT ?':NEWKEY://
'ARE YOU SAYING NO JUST TO BE NEGATIVE?':NEWKEY://
'YOU ARE BEING RATHER NEGATIVE':NEWKEY://
MODROP L /20/ D //
'THE MODROP COMMAND PREVENTS THE SYSTEM FROM RELEASING CURRENT WORKING FILES WHEN THE USER ISSUES THE OLD, NEW, OR LIB COMMAND TO OBTAIN A NEW PRIMARY FILE. THIS COMMAND MUST BE ENTERED IMMEDIATELY AFTER THE OLD, NEW, OR LIB COMMAND SEQUENCE IS COMPLETE':CF SPEC:NEWKEY://
NR L /21/ D //

'NR REFERS TO THE NO REWIND OPTION. IF PRESENT, THIS PARAMETER WILL INHIBIT REWIND AFTER EXECUTION OF ASSOCIATED COMMAND'.NEWKEY:NEWKEY://

OFF L /20/ D //CF LOGOFF://

DFN L /21/ D ///DFN REPRESENTS OLD FILE NAME IN CHANGE COMMAND':

OLD L /20/ D //

'THE OLD COMMAND RETRIEVES A COPY OF THE SPECIFIED PERMANENT FILE (INDIRECT) FOR USE AS THE PRIMARY FILE'.THE FORM OF THE OLD COMMAND IS - OLD;LFN=PFN/UN=USERNUM;PW=PASSWRD://

'CF SPEC:NEWKEY://

OPERATING L /18/ D //CF KRONOS://

OPTIONS L /19/ D //CF LO://

P L /21/ D //CF PRIVATE://

PARAMETER L /19/ D //CF SPEC://

PARAMETERS L /19/ D //CF SPEC://

PASSWORD L /20/ D //CF PW://

PASSWRD L /20/ D //CF PW://

PERHAPS L /2/ D //

'YOU DONT SEEM TO BE QUITE CERTAIN.':'WHY THE UNCERTAIN TONE?':

'CANT YOU BE MORE DEFINITE?':'YOU ARENT SURE?':'DON'T YOU KNOW?':

PERMANENT L /18/ D //CF FILES://

PERMIT L /20/ D //

'THE PERMIT COMMAND IS USED TO GRANT PERMISSION FOR A USER UNDER A SPECIFIED NUMBER TO ACCESS A PRIVATE FILE'.THE FORM OF THE PERMIT COMMAND IS - PERMIT,PFN,USERNUM1=M1,USERNUM2=M2,...,USERNUMN=MN/R=R. WHERE THE M IN EACH CASE REPRESENTS THE PERMISSION MODE':CF SPEC:NEWKEY://

PFILE L /18/ D //CF FILE://

PFFILES L /18/ D //CF FILE://

PFN L /21/ D ///PFN REPRESENTS PERMANENT FILE NAME ':/

PRGCOPY L /20/ D //

'PRGCOPY IS A ROUTINE WHICH ALLOWS THE USER TO RETRIEVE A PERMANENT FILE WHICH HAS BEEN PURGED IN ACCORDANCE WITH THE PFILE PURGING ALGORITHM'.THE USER MAY ACCESS THE PRGCOPY ROUTINE BY THE COMMAND PRGCOPY,DATE,PFNNAME. WHERE DATE REPRESENTS THE NUMERIC DAY OF THE MONTH WHEN THE FILE WAS PURGED; AND PFNAME REPRESENTS THE NAME OF THE PFILE WHICH IS TO BE RECOVERED. THE DATE IS A REQUIRED PARAMETER AND IS AVAILABLE FROM PRGLIST. IF PFNAME IS OMITTED, ALL OF THE USERS PFILES ARE RECOVERED FROM THAT DAYS PURGE TAPE. NOTE THAT PFILES MAY NOT BE RECOVERED AFTER ONE MONTH. ALSO NOTE THAT, TO ACCOMPLISH THIS FEAT FROM THE TELETYPE, A SUBMIT COMMAND MUST BE USED':CF SPEC:NEWKEY://

PRGLIST L /20/ D //

'PRGLIST IS A ROUTINE WHICH ALLOWS THE USER TO FIND OUT WHICH OF HIS PFILES HAS BEEN PURGED AND WHICH OF THEM IS TO BE PURGED SOON':

PRGLIST MAY BE USED IN ANY OF THE FOLLOWING THREE FORMS

PRGLIST - WHICH LISTS FILES PURGED TODAY AND THOSE SCHEDULED FOR TOMORROW,

PRGLIST,DATE - (WHERE DATE IS OF THE FORM YR/MO/DY) WHICH LISTS ALL PFILES PURGED SINCE THE DATE SPECIFIED, AND PRGLIST,, - WHICH LISTS ALL FILES PURGED IN THE PAST MONTH':CF SPEC:NEWKEY://
PRIMARY L /21/ D //

'THE PRIMARY FILE IS ONE TYPE OF WORKING FILE. IT HAS SPECIAL
SIGNIFICANCE IN CERTAIN TIME-SHARING COMMANDS. A PRIMARY FILE IS OBTAINED
WITH THE OLD OR LIBRARY COMMAND WHICH RETRIEVES A COPY OF AN INDIRECT
ACCESS PERMANENT FILE. A PRIMARY FILE IS CREATED WITH THE NEW COMMAND.
THERE IS ONLY ONE PRIMARY FILE ACTIVE OR AVAILABLE TO THE USER AT ANY
GIVEN TIME':CF SPEC:NEWKEY:/

PRIVATE L /20/ D //

'P REPRESENTS A PRIVATE FILE. THESE FILES ARE AVAILABLE
FOR ACCESS ONLY BY THE ORIGINATING USER OR BY THOSE EXPLICITLY GRANTED
PERMISSION (REFER TO PERMIT COMMAND)':/

PUBLIC L /20/ D //

'PU REPRESENTS A PUBLIC FILE. THESE FILES ARE AVAILABLE FOR ACCESS
BY ALL USERS KNOWING THE FILE NAME, PASSWORD, AND USER NUMBER. ONLY
THE TOTAL NUMBER OF ACCESSES TO FILES OF THIS CATEGORY ARE RECORDED
FOR THE ORIGINATOR OF THE FILE':/

PURGE L /20/ D //

'THE PURGE COMMAND REMOVES THE SPECIFIED PERMANENT FILE FROM STOPAGE.
FILES REMOVED IN SUCH A MANNER CANNOT BE RECOVERED BY PRGCDPY':/
'THE FORM OF THE PURGE COMMAND IS -
PURGE,FFN/UN=USERNUM,
PW=PASSWORD':CF SPEC:NEWKEY:/

PURGED L /19/ D //

'DO YOU WANT TO KNOW ABOUT PRGLIST, WHICH GIVES A LISTING OF YOUR
PURGED FILES, OR DO YOU WANT TO KNOW ABOUT PRGCDPY, WHICH WILL ALLOW
YOU TO RETRIEVE A COPY OF A PURGED FILE?':CF SPEC:NEWKEY:/
PW=PASSWORD L /21/ D //CF PW:/'
PW=PASSWORD L /21/ D //CF PW:/'
PW L /21/ D //

'PW REPRESENTS A PASSWORD. THE USER HAS THE OPTION OF SPECIFYING A
ONE-TO-SEVEN CHARACTER PASSWORD FOR A FILE. THIS PASSWORD MUST BE
SPECIFIED WHENEVER ALTERNATE USERS ACCESS THE FILE':/

Q L /21/ D //

'Q REPRESENTS THE QUEUE SPECIFIED IN A SUBMIT COMMAND. IT MAY BE
EITHER B, N, OR E. B MEANS THAT THE JOB OUTPUT IS DISPOSED TO THE BATCH
QUEUE TO BE PRINTED, PUNCHED, ETC., AT WCC. N MEANS THAT THE JOB OUTPUT
IS DISPOSED TO THE BATCH QUEUE, BUT IS DROPPED AT JOB TERMINATION.
DEFAULT VALUE). E MEANS THAT THE JOB OUTPUT IS DISPOSED TO EXPORT/IMPORT
QUEUE FOR PRINTING AT A REMOTE SITE':NEWKEY:NEWKEY:/

R=R L /21/ D //CF R:/'
R L /21/ D //

'R, IF PRESENT IN THE LIST COMMAND, INDICATES THAT END-OF-RECORD
AND END-OF-FILE MARKS ARE TO BE INDICATED IN THE LISTING IF PRESENT':
NEWKEY:NEWKEY:NEWKEY://

READ L /21/ D //READ MODE/
' R REPRESENTS THE READ MODE AND ALLOWS A USER TO READ AND/OR EXECUTE
THE FILE':CF SPEC:NEWKEY:/
RA L /21/ D //CF READAP://
READAP L /21/ D //

'READAP represents the READ/APPEND mode and allows the user to read a
DIRECT ACCESS file with the implication that another user may currently
be accessing the file in APPEND mode. The file may also be executed in
this mode':CF SPEC:NEWKEY://

RM L /21/ D //CF REAMD://

'REAMD represents the READ/APPEND mode and allows the user to read a
DIRECT ACCESS file with the implication that another user may currently
be accessing the file in APPEND mode. The file may also be executed in
this mode':CF SPEC:NEWKEY://

READM L /21/ D //CF READMD://

'READM represents the READ/MODIFY mode and allows the user to read a
DIRECT ACCESS file with the implication that another user may currently
be accessing the file in MODIFY mode. The file may also be executed in
this mode':CF SPEC:NEWKEY://

RECORD L /19/ D //CF FILE://

RECORDS L /19/ D //CF FILE://

RECOVER L /20/ D //

'The RECOVER feature enables the user at a time-sharing terminal to
resume processing after having been accidentally disconnected from the
system or when a system malfunction requires that the log-in sequence
be reinitialized. The user is placed in recovery state whenever he is
disconnected from the system without logging off, providing that he is
not already in recovery state': RECOVERY MUST BE INITIATED WITHIN 10
MINUTES OF BEING DISCONNECTED. THIS IS DONE IN THE LOGIN SEQUENCE. IN
RESPONSE TO THE PROMPT RECOVER/SYSTEM THE USER ENTERS RECOVER,NNN
WHERE NNN REFERS TO THE TERMINAL BEING USED WHEN THE FAILURE OCCURRED.

THIS IS THE SAME NUMBER INDICATED WHEN THE USER INITIALLY LOGGED IN.
If the same terminal number is indicated when the user logs in to recover,
THIS PARAMETER IS NOT REQUIRED':CF SPEC:NEWKEY://

RECOVERY L /20/ D //CF RECOVER://

RELEASE L /20/ D //CF RETURN://

REPLACE L /20/ D //

'The REPLACE command allows a user to replace the contents of a
PERMANENT file (PFN) with the contents of a working file (LFN)'
THE FORM OF THE REPLACE COMMAND IS - 
REPLACE,LFN=PFN/UN=
USERNUM,PW=PASSWORD':CF SPEC:NEWKEY://

RETRIEVE L /18/ D //CF GET://

'RETURN L /20/ D //

'TO RELEASE WORKING FILE LFN, ENTER RETURN,LFN ': NEWKEY:NEWKEY://

REWIND L /20/ D //

'TO POSITION WORKING FILE LFN AT THE BEGINNING-OF-INFORMATION
(BOI) ENTER REWIND,LFN ':NEWKEY:NEWKEY://

ROUTE L /20/ D //

'The ROUTE command can be used to send a file to a specific location or
piece of equipment or to make multiple copies of the same file'
THE FORM OF THE ROUTE COMMAND IS - 
ROUTE(LFN,EQ,C=CC,
L=LOC,EQ=EQ,AT=AT)':CF SPEC:NEWKEY://

SAVE L /20/ D //

'The SAVE command creates an indirect access permanent file. This
command allows a user to retain a copy of the specified working file in the
permanent file system': THE FORM OF THE SAVE COMMAND IS - 
SAVE,LFN=PFN/PW=PASSWORD,CT=N,M=M':CF SPEC:NEWKEY://
SEMI-PRIVATE represents a semi-private file. These files are available for access by all users knowing the file name, password, and user number. Accesses by alternate users for files of this category are recorded for the originator of the file. This includes the user number of the alternate user, the number of accesses made, and the date and time of the last access (refer to CATLIST command). 

```
THE STATUS COMMAND REQUESTS THE CURRENT JOB STATUS. AN IMPORTANT FEATURE OF THIS COMMAND IS THAT IT MAY BE ENTERED DURING JOB EXECUTION:
```

```
'BESIDES THE SIMPLE STATUS COMMAND WITH NO PARAMETERS, THERE ARE FOUR ALTERNATE COMMANDS -

STATUS,L=LOC STATUS,F STATUS,T STATUS,J=JOBNAME

STATUS,L=LOC STATUS,F IS THE SAME AS STATUS BUT ALSO LISTS ALL WORKING FILES;
STATUS,T REQUESTS THE ACCUMULATED CPU TIME FOR THIS SESSION;
STATUS,J=JOBNAME REQUESTS THE STATUS OF A REMOTE BATCH JOB OR JOB SUBMITTED AT WCC, BALANTINE, BUSINESS, OR MEMORIAL; AND STATUS,L=LOC REQUESTS THE JOB STATUS OF ALL JOBS AT LOCATION LOC':
```

```
THE STOP COMMAND TERMINATES ANY PROGRAM THAT IS CURRENTLY EXECUTING OR WAITING FOR INPUT FROM THE TERMINAL:
```

```
THE SUBMIT COMMAND ALLOWS A VALIDATED USER TO CREATE A BATCH JOB DECK AND SUBMIT IT TO THE LOCAL BATCH QUEUE AS A DEFERRED BATCH JOB. THE USER CANNOT INTERACT WITH A JOB SUBMITTED FOR PROCESSING USING THIS COMMAND. RATHER, THE JOB IS TREATED IN THE SAME MANNER AS A JOB SUBMITTED FROM THE CARD READER AT WCC, BALANTINE, MEMORIAL, OR BUSINESS:
```

```
THE FORM OF THE SUBMIT COMMAND IS -
```

```
SYSTEM L/18/ D/CF KRONOS/;
```

```
TELEX L/17/ D/CF KRONOS/:
```

```
TELEX IS THAT PART OF KRONOS WHICH DEALS WITH INTERACTIVE USERS:
```

```
UN=USERNUM L/21/ D/CF UN/;
```

```
UN=USERNUM L/21/ D/CF UN/;
```

```
UN REPRESENTS A FOUR-DIGIT USER NUMBER:
```

```
WCC L/19/ D/CF WCC/;
```

```
'DO YOU WANT TO SUBMIT, ROUTE, OR DROP A JOB AT WCC?':
```

```
WONT S/1/ WONT/;
```

```
Appendix III: Cal SNØBØL Version of ELIZA

AN EXPLANATION OF CRYPTIC PATTERNS:

- P.1 BREAKS A SENTENCE AT WORD BOUNDARIES
- P.3 BREAKS CONTENT FROM A STRING OF CUES OR A STRING IN MEMORY
- P.4 SHORTENS STRINGS BY PRE AND POST TRIMMING
- P.5 MATCHES 'S', 'L', OR 'D' INDICATOR
- P.6 MATCHES SPECIAL INDICATOR
- P.7 MATCHES ASSOCIATED WORD IN MEMORY
- P.8 EXTRACTS SENTENCE FROM WITHIN MEMORY QUEUE
- P.9 EXTRACTS SENTENCE FROM FRONT OF QUEUE

FUNCTION FOR FINDING INTEGERS...

```
DEFINE('INTEGER(I)') :<INTOUT>
```

```
INTEGER I POS(0) SPAN('0123456789') RPOS(0) \s<RETURN> F<RETURN>
INTOUT ANCHOR(1)
```

```
DEFINE INPUT FILES...
DETACH('INPUT')
INPUT('INPUT', 'INPUT', 80)
INPUT('FILEA', 'SCRIPTH', 80)
```

```
DEFINE OUTPUT FILE...
OUTPUT('SAVER', 'ROUTIT')
```

```
PATTERNS AND OTHER MEMORABILIA...
PRE.TRIM = SPAN(' ') \sNULL
THANKS = FENCE 'THANK' (SPAN(' ') 'YOU' \s'S')
QUEST.TRIM = RTAB(1) \sPHRASE. '?'
P.1 = PRE.TRIM BREAK(' ') \sWORD.
P.3 = BREAK('\s') \sCONTENT. ':'
P.4 = PRE.TRIM REM \sLESS.
P.5 = PRE.TRIM ANY('SLD') \sWORD.
P.6 = PRE.TRIM ANY('ABCEFHL') \sBRANCH.
P.7 = SPAN(' ') \sWORD. SPAN('')
P.8 = (ARBNO(BREAK('') '') \sSTR1. (ARBNO(NOTANY(''))) \sSPAN('') \sWORD. SPAN('') BREAK('') \sCONTENT. ':' \sREM \sSTR2.
P.9 = (BREAK('') \sOUTPUT \sSAVER 'i') \sSTR1. REM \sSTR2. OUTP = (LEN(SO) BREAK('') SPAN('')) \sOUTPUT \sSAVER SHORTEM = BREAK('') \sPHRASE. ANY('\s') \sREM \sTRAILER.
X.REF = PRE.TRIM FENCE 'CF'
BUMP. = PRE.TRIM 'NEWKEY'
PAREN. = PRE.TRIM 'B Break('')', CONTENT. ''
CALL.TO.SNOBOL = PRE.TRIM 'SNOBOL'
```

```
FAM1 = 'MOTHER' \s'FATHER' \s'SISTER' \s'BROTHER' \s'DAUGHTER'
FAM2 = 'MOM' \s'DAD' \s'WIFE' \s'CHILDREN' \s'HUSBAND' \s'SON'
FAMILY = <FAM1 \sFAM2> \sRELATIVE
BELIEF = ('FEEL' \s'THINK' \s'BELIEVE' \s'WISH') \sFENCE
HIGH = ('HAPPY' \s'ELATED' \s'GLAD' \s'BETTER' \s'HIGH') \sBIEN
MULTI = ('EVERYONE' \s'EVERYBODY' \s'NOBODY' \s'NOONE') \sALLES
ICKY = ('SAID' \s'UNHAPPY' \s'DEPRESSED' \s'SICK') \sMALADE
```
IMPORTANT STRINGS...

INTRODUCTION = 'HOW DO YOU DO. WHAT IS YOUR PROBLEM?'
CLUELESS = '.. VERY INTERESTING'
:WOUID YOU MIND REPEATING THAT?'
'I DONT SEEM TO UNDERSTAND WHAT YOU ARE SAYING'
'COULD YOU CLARIFY THAT STATEMENT PLEASE?'
RETAI = 'MY.YOUR.I.YOU.'

WE NOW READ THE SCRIPT AND FORM STRINGS AS FOLLOWS...

FOR EACH KEY WORD 'xxxx' WE FORM THE FOLLOWING VARIABLES:

RPL.xxxx IS A REPLACEMENT WORD. (OPTIONAL)
LEV.xxxx IS A LEVEL NUMBER (IF ABSENT KEY IS IGNORED)
N.xxxx A COUNT OF THE NUMBER OF DECOMPOSITIONS
DEC.I.xxxx IS THE I' TH DECOMPOSITION PATTERN
RULE.I.xxxx IS A STRING OF DECOMPOSITION RULES FOR THE I' TH

DECOMPOSITION. RULES ARE SEPARATED BY ':'

KEYWORDS. = '

READ IN SCRIPT...
HIGGINS SCRIPT. = TRIM(FILEA)
IDENTSCRIPT. 'END') :S(INTRO)

EXTRACT KEY WORD FROM SCRIPT...
SCRIPT.P.1 = :F(HIGGINS)

PLACE KEY WORD ON LIST OF KEY WORDS IF APPROPRIATE...
KEY. = ' ' WORD.
KEYWORDS. KEY. '
':S(LESSON)
KEYWORDS. = KEY. KEYWORDS.

EXTRACT 'S', 'L', OR 'D' AND BRANCH ACCORDINGLY...

LESSON SCRIPT. P.5 = :F(HIGGINS)$($WORD.)

ERROR OUTPUT = 'SCRIPT ERROR: ' WORD. ' SCRIPT. :F(HIGGINS)

SUBSTITUTION RULE - EXTRACT STRING AND STORE...
S STORE. = 'RPL' KEY.
'SCRIPT. PRE.TRIM ' BREAK(') . $STORE. '/' = :F(ERR)
':F(LESSON)

LEVEL NUMBER - EXTRACT STRING, CHECK IF NUMBER, AND STORE...
L SCRIPT. PAREN. = :F(ERR)

$:('LEV' KEY.) = INTEGER(CONTENT.) CONTENT. :F(LESSON)

DECOMPOSITION - SET UP DECOMPOSITION NUMBER AND PATTERN...
D N.M = $(N' KEY.) + 1
$(N' KEY.) = N.M
SCRIPT. PAREN. = :F(ERR)

CHECK IF SPECIAL RULE IS TO BE STORED AND BRANCH WHERE NECESSARY...

CONTENT. CALL.TO.SPECIAL = :S(SPECIAL)
$:('DEC.' N.M KEY.) = ARB CONTENT. REM. POST
 RULES STORE. = 'RULE.' N.N KEY.
 $STORE. = DIFREP (SCRIPT.) SCRIPT. 
 :F (NEW.CARD)
 LOOP $STORE. RTAB (1) / / :S (HIGGINS)
 NEW.CARD IHOLD = TRIM (FILEA)
 IDENT (IHOLD, 'END') :S (INTRO)
 $STORE. = $STORE. IHOLD 
 (LOOP)

+ THE FOLLOWING ARE SPECIAL SCRIPT-HANDLING STATEMENTS

SPECIAL CONTENT. P. 6 = :F (ERR) $S (BRANCH.)

'ERR1 OUTPUT = 'SCRIPT ERROR: ' BRANCH. :S (HIGGINS)
 A $(<DEC. N.N KEY.) = ARB MULTI REM . POST : (RULES)
 B $(<DEC. N.N KEY.) = ARB 'YOU' 'WANT' 'NEED' REM . POST
 . : (RULES)
 C $(<DEC. N.N KEY.) = ARB 'YOU ARE' ARB ICKY : (RULES)
 D $(<DEC. N.N KEY.) = ARB 'YOU ARE' ARB HIGH : (RULES)
 E $(<DEC. N.N KEY.) = ARB 'BELIEF' 'YOU' REM . POST
 . : (RULES)
 F $(<DEC. N.N KEY.) = ARB 'CANNOT' 'CANT' REM . POST
 . : (RULES)
 G $(<DEC. N.N KEY.) = ARB 'YOU' 'IS' 'ARE' 'WAS' 
 . ARB 'LIKE' : (RULES)
 H $(<DEC. N.N KEY.) = ARB 'YOU' ARB . POST 'I'
 . : (RULES)
 I $(<DEC. N.N KEY.) = ARB 'AM' 'IS' 'ARE' 'WAS'
 J $(<DEC. N.N KEY.) = ARB 'FAMILY' REM . POST : (RULES)
 K $(<DEC. N.N KEY.) = ARB 'I' ARB . POST 'YOU' : (RULES)

+ WE NOW HOLD A CONVERSATION. FIRST WE READ A SENTENCE AND
SEARCH FOR KEY WORDS REPLACING APPROPRIATE ONES
AND STACKING THE KEYS IN A QUASI-ORDERED LIST (STRINGS).

INTRO OUTPUT = 'A COPY OF THIS CONVERSATION IS BEING'
OUTPUT = 'MADE FOR FURTHER STUDY...'
OUTPUT = 'TO TERMINATE THIS PROGRAM TYPE QUIT,'
OUTPUT = 'THANKS, OR THANK YOU...'
OUTPUT = INTRODUCTION
SAVER = INTRODUCTION

+ GET INPUT STRING...

HEAR PHRASE. = TRIM (INPUT) 
 :F (END)
SAVER =
SAVER = '***INPUT: ' PHRASE.
SAVER =
PHRASE. QUEST. TRIM
PHRASE. = PHRASE. '/'
IDENT (PHRASE. 'QUIT.' ) :S (END)
IDENT (PHRASE. 'REPUN.' ) :S (INTRO)
PHRASE. THANKS :S (NICE.END)
ANCHOR (1)
LOOKBACK =
LOOK.B =
ASS. FLAG =
• SHORTEN INPUT STRING WHEN APPROPRIATE...

HEARLESS PHRASE. SHORTEN.
PHRASE. = PHRASE. ' '
IMAGE. =

• REMEMBER CUES FROM PREVIOUS SENTENCE, INCLUDING KEY CUE ONLY IF NECESSARY.
OLD.CUES = DIFFER(CUES.) CUES. ;S(OMIT.CUE)
(CUE. ':;' DIFFER(CUE. ) ':;' (RTAB(1) ':;' ) . OLD.CUES

OMIT.CUE CUES. =
CUE.LEVEL = 0
ANCHOR()

• GET WORD.
SPLIT PHRASE. P.1 =
F(REPLY1)

• CHECK IF WORD IS A KEYWORD.
KEYWORDS. ':;' WORD. ':;'
F(KEEP)

• MAKE SUBSTITUTION IF REQUIRED.
NEW.WORD = S('<RPL.' WORD. )
IMAGE. = DIFFER(TPIM(NEW.WORD) IMAGE. NEW.WORD) ;S(STACK)
IMAGE. = IMAGE. WORD. ':;'

• NOTHING ELSE IS DONE IF NO LEVEL NUMBER
STACK NEW.LEVEL = DIFFER($('<LEV.' WORD. ))

IMAGE. = $('<LEV.' WORD. )
CUE.LEVEL = GT(NEW.LEVEL,CUE.LEVEL)

NEW.LEVEL
CUES. = WORD. ':; CUES. ;S(SPLIT)
LOCUE CUES. = CUES. WORD. ':; ;S(SPLIT)
KEEP IMAGE. = IMAGE. WORD. ':; ;S(SPLIT)

• CHECK MEMORY FOR ASSOCIATION:
MEMORY. GT(SIZE(WORD.),4) P.7
ASS.FLAG = 'YES'
MEMORY. P.8 = STR1. STR2. ;F(ERR3) $<SPLIT)

ERR3 OUTPUT = 'ERROR IN PATTERN P.8:
MEM OUTPUT = 'MEMORY:
OUTPUT = MEM.
S(SPLIT)

• THIS PART FORMS OUR REPLY TO THE INPUT SENTENCE

REPLY1 IDENT(ASS.FLAG) ;F(ASSOC) $<REPLY
NEWCUE CONTENT. P.4 = TRIM(LESS.) ;<NEXTCUE
REPLY CUES. P.3 = ;<NOCUE
NEXTCUE CUE. = ':; CONTENT.
N.N = 0
NMAX. = $('<N' CUE. )
ANALYSE N.N = LT(N.N,NMAX.) N.N + 1
IMAGE. $('<DEC.' N.N CUE. )
$('<RULE.' N.N CUE. ) P.3 =
CONTENT. POS(0) ':; =
$('<RULE.' N.N CUE. ) =$('<RULE.' N.N CUE. ) CONTENT. ':;'
CONTENT. X.REF =
CONTENT. BUMP.
S(REPLY)
• THE RECOMPOSITION RULES ARE JOINED WITH THE PATTERN AND PUT TO OUTPUT

ANCHOR()
BEFORE =
AFTER =
HOLD =

• DELETE LEADING AND TRAILING BLANKS FROM CONTENT...
  CONTENT, P.4 = TRIM(LESS).

• DECOMPOSITION RULES MUST BE QUOTED...
  CONTENT, POS(0) ("\"\"") , QUOD =
  ERR2 OUTPUT = "ERROR IN RECOMPOSITION RULE / CONTENT."
  (HEAR)

• DEAL WITH UNQUOTED PARTS OF THE DECOMPOSITION RULE...
  CONTENT, BREAK(QUOD) , BEFORE QUOD =
  CONTENT, BREAK("\"\"\") , HOLD, = DIFFER(HOLD.) :F(OUT)
  CONTENT, ("\"\"\") $ QUOD PTAB(1) , AFTER QUOD =

• CLEAN HOLD, FOR INVERSION IF NEEDED...
  HOLD, P.4 = TRIM(LESS.)

• ARE WE CURRENTLY LOOKING BACK TO A PREVIOUS SENTENCE?
  LOOK.B = DIFFER(LOOKBACK) OLD.HOLD

• REMEMBER HOLD, STRING IF NEEDED FOR LOOKING BACK LATER...
  OLD.HOLD = DIFFER(HOLD.) $HOLD, :F(NO,HOLD)

• IF LOOKING BACK, RESET VALUE OF $HOLD....
  $HOLD, = DIFFER(LOOKBACK) LOOK.B

• AN ANSWER ESCAPES...
  OUTS, = DIFFER(HOLD.) BEFORE $HOLD, AFTER :S(PRINT)
  NO.HOLD OUTS, = BEFORE AFTER

PRINT OUTS, OUTPUT =
  OUTPUT = OUTS,
  SAYER = OUTS,

SETA ANCHOR(0)

• DOES ELIZA WANT TO REMEMBER THIS SENTENCE?
  RETAIN IDENT(LOOKBACK) CUE.
  :F(HEAR)
  MEMORY, = LT(SIZE(MEMORY.) , 200) MEMORY, IMAGE, / : / (HEAR)

• THIS IS WHAT WE DO IF THERE ARE NO KEY WORDS IN THE INPUT

• DO WE HAVE MORE OF THE INPUT SENTENCE TO CONSIDER...

NO.CUE PHRASE, = DIFFER(TRAILER.) TRAILER, :S(HEARLESS)

• IF NEEDED, REVIVE CUES FROM PREVIOUS SENTENCE...
  CUES, = DIFFER(OLD.CUES) OLD.CUES :F(RECALL)
  OLD.CUES =
  LOOKBACK = "YES" : (REPLY)

• LOOK FOR EARLIER UPEC FROM MEMORY QUEUE...

RECALL MEMORY, P,3 =
  OUTS, = ".. EARLIER YOU SAID / CONTENT."
  :F(ER,AE,UM)

• MAKE ASSOCIATION WITH EARLIER WORD...

ASSOC OUTS, = "DOES THAT HAVE ANYTHING TO DO WITH THE FACT /
  \ THAT / TRIM(CONTENT.) / ?" : (PRINT)
EVERYTHING HAS FAILED AT THIS POINT AND ELIZA STAMMERS...
ER.AH.UM CLUELESS P.9 = STR2. STR1.   :(HEAR)
NICE.END OUTPUT = 'MY PLEASURE'
SAVER = 'MY PLEASURE'

AN EXPLANATION OF THE ORGANIZATION OF SCRIPT FILES:

THE FIRST LINE OF A FILE ENTRY CONTAINS THE KEY WORD
FOLLOWED BY AN 'L' AND ITS LEVEL NUMBER ENCLOSED IN SLASHES
FOLLOWED BY AN 'S' AND ITS SUBSTITUTION STRING ENCLOSED IN
SLASHES, FOLLOWED BY A 'D' AND THE STRING TO BE USED IN THE
DECOMPOSITION PATTERN. (EACH OF THESE PARTS ARE OPTIONAL AND
THE ORDER IN WHICH THEY APPEAR IS NOT IMPORTANT EXCEPT THAT THE
KEY WORD MUST APPEAR FIRST.)

SUBSEQUENT LINES OF SCRIPT MAY CONTAIN ANY NUMBER OF
DECOMPOSITION RULES ENCLOSED IN SINGLE QUOTES (' ') AND DELIMITED
BY COLONS (:) ENDING IN COLON SLASH (:/). (THIS PART IS ALSO
OPTIONAL.)

WITHIN EACH RULE, THE VARIABLE POST MAY APPEAR UNQUOTED
WHEREVER THE SUBSTRING OF THE INPUT SENTENCE (REMAINING AFTER THE
DECOMPOSITION PATTERN WAS APPLIED) IS TO APPEAR IN THE OUTPUT
SENTENCE.

TWO OTHER METHODS OF SPECIFYING A DECOMPOSITION RULE MAY BE
USED. ONE WAY IS TO SIMPLY USE THE WORD NEWKEY (UNQUOTED)
AND THE ASSOCIATED KEY WORD WILL BE IGNORED. A NEW KEY WORD
WILL BE TAKEN FROM THE CUES, STACK AND SUBSEQUENT ACTION
PERFORMED. THE SECOND WAY IS TO USE THE LETTERS CF (UNQUOTED)
FOLLOWED BY ANOTHER KEY WORD SYNONYMOUS WITH THIS ONE. THUS,
WORDS CAN BE IGNORED EXCEPT FOR THEIR SUBSTITUTIONS,
WORDS CAN EVOKE THE EXACT RESPONSES OF ANOTHER WORD, OR
WORDS CAN TRIGGER ORIGINAL RESPONSES OCCASIONALLY AND BE IGNORED
THE REST OF THE TIME.

END