

APPENDIX E: WYLBUR REFERENCE MANUAL

Second Edition

February 29, 1968

Campus Facility USERS MANUAL



PREFACE

This second edition of the WYLBUR users manual updates the first edition to reflect the new features which have been added to WYLBUR since its introduction to the Stanford University Community. The developers of the WYLBUR system wish to offer their thanks to the early users who suffered through the many system crashes which occurred while producing the stable system enjoyed today.

WYLBUR was produced by the programming staff of the Stanford Computation Center, Campus Facility. Credits are due all of the Staff of the Campus Facility without whose individual efforts WYLBUR could not have been born or reached its young manhood. Warranting particular mention are Roger Fajman, John Borgelt, John Gilman, Leighton Allen, Janet Gallo, Jim Moore and Bill Riddle upon whose shoulders the bulk of the design, implementation, and documentation fell. We also wish to express a special word of thanks to the Director of the Stanford Computation Center, Professor E. A. Feigenbaum, for his support during the development of the system.

WYLBUR is a subsystem residing under what is called the Stanford Terminal Processor (STP) and is the first of many terminal oriented facilities we hope to develop here at Stanford.

The design goal was to produce a terminal oriented text editor and remote job entry facility with fairly strong data management facilities to be used as an aid in program development. WYLBUR is not a time sharing facility in the sense that that word means interactive computation. It is, however, a system which provides quick response to many users simultaneously sitting at terminals doing text editing.

Additional power is provided through the integration of STP with HASP which is the spooling submonitor used to control the batch services here at Stanford. Modification to this piece of work provides for the low overhead introduction of programs into the batch stream, in an easy and natural way, from the WYLBUR terminal. Output from

E- 2

batch stream programs can be left on disk for later review from the terminal. While this does not provide time sharing in the formal sense, it does provide a more favorable facility for getting work done than that which can be provided by a batch system alone.

Rod Fredrickson
Associate Director,
Stanford Computation Center,
Campus Facility

W Y L B U R

Stanford University
Computation Center
Text Editor

by William E. Riddle

WYLBUR is a Text Editor designed to operate in either an O/S or HASP environment in order to provide users at 2741 terminals a comprehensive text editing facility with prompt response, while otherwise allowing the computer to proceed with processing of the normal batch job stream.

WYLBUR allows the user to change the contents of a line without retyping all of it, to insert, delete or replace lines, to copy and move lines, to renumber lines, and to list lines. In addition, the user may retrieve external data sets in order to work upon them and save data sets which have been constructed. The user may also insert the data set into the monitor's input stream for processing as a batch job.

This manual is intended to serve a dual purpose. First it should serve as a planning guide for potential users -- an introduction to what the user can and cannot accomplish using WYLBUR. Second, it should serve the user while he is operating WYLBUR -- a reference text.

The format of the manual is as follows. In order to

make the later discussion more readable, the definitions of all terms are gathered into an initial section. The remaining sections introduce the WYLBUR commands in order of their increasing difficulty. In introducing the commands, the form of the manual is conversational with examples given showing what will happen during a session with WYLBUR when the command under discussion is used. In the examples, all material which is typed at the terminal by WYLBUR is underlined to differentiate it from the typing done by the user. Also, the typing of the non-printing keys ATTN and RETURN are designated by ATTN and CR.

DEFINITION OF BASIC TERMS

Before outlining the set of operations which is available in WYLBUR, it is necessary to define some basic terms which will be needed in the discussion.

The elementary "atomic unit" from which all data sets are ultimately derived is the CHARACTER. The set of valid WYLBUR characters - the WYLBUR CHARACTER SET - consists of those on the keyboard of an IBM 2741 teletypewriter, a picture of which appears in Appendix A. This is a standard typewriter keyboard with a few special purpose keys, the function of which will be explained shortly, added.

The 2741 contains some characters which are not available on the 1403 high speed printers, ! and ¢. These extra characters will be printed as the blank character when a WYLBUR created data set is printed on the 1403.

On a 2741 keyboard there is also a set of keys which do not cause any printing to occur. These include the special-purpose keys which were mentioned previously. Some of these have a special meaning when used under WYLBUR and hence a full description of their function is deferred until the WYLBUR commands are explained. However, a rough description of the function of these NON-TYPING KEYS is given now in order to complete the discussion of the WYLBUR character set.

RETURN - this key returns the printing head to the left-hand margin. In WYLBUR it is used during a session to signal that the user has finished typing a line and wishes WYLBUR to inspect the line and effect any operations which are called for.

BACK-SPACE - this key moves the printing head one space to the left. In WYLBUR this effectively erases this character from the line of text which the user is currently typing.

TAB - this moves the print head forward to the next tab setting. The user must follow a set of very rigid rules, which will be explained subsequently, concerning its use.

CLR and SET toggle switch - this is used to clear and set the tabs. There are also rigid rules concerning the use of this switch which will be explained below.

MAR REL - margin release key used as on a standard typewriter. This key serves no purpose whatsoever in WYLBUR.

SHIFT and LOCK - used to shift into upper case as on a standard typewriter.

ON and OFF toggle switch - since the 2741 is basically an electric typewriter, this switch serves the function of turning on and off the terminal. It should not be used during a WYLBUR session since the results might be disastrous.

ATTN - used during a session to tell WYLBUR that it should suspend whatever it is currently doing for the user and prepare to accept a new directive.

It should be reiterated that the above explanation of the RETURN, ATTN, and BACK-SPACE keys is inadequate for a full understanding of their function in WYLBUR. The full story on the use of these keys will unfold as the various WYLBUR commands are explained, and then be brought together in a succinct outline in the appendix.

Characters, the elementary building blocks, are used to construct the following increasing hierarchy of forms.

Character String - any group of consecutive characters which occurs on one line.

Line - a character string which begins at the left margin and ends with a carriage return.

Range - a group of lines, each of which possesses some distinguishing characteristic.

Data Set - a group of lines which, taken as a whole, forms some logical unit of interest.

The rest of this section is devoted to a fuller description of each of these constructs, their forms and attributes, and their functions in WYLBUR.

CHARACTER STRINGS are the basic unit of interest in WYLBUR since it is these entities which a WYLBUR user generally wishes to manipulate. A character string is represented by a group (of at least one) of consecutive characters from the WYLBUR character set. Blanks are significant - a group of three (say) blanks appears in the string as three blank characters. The significance of blanks also applies to leading (those before any other characters in the string) and terminal (those following all

other characters in the string) blanks.

A LINE is a special character string which begins at the left-hand margin and continues to a carriage return. It is also differentiated from a normal character string in that it forms a logical unit. Two types of lines may be distinguished by the type of logical unit they form. The first is a COMMAND LINE which has the following format

<WYLBUR REQUEST> <USER RESPONSE> CR

Essentially, WYLBUR asks the user a question by typing the <WYLBUR REQUEST> prompt at the user's terminal. Then the user completes the line by typing his <USER RESPONSE> answer. As an example, WYLBUR may ask the user a question and then, on the next line, prompt the user for an answer to the question that has been posed

REPLY ?

The user answers by completing the line with his answer and then finishing the line with a carriage return

REPLY ? YES CR

The second type of line is a LINE OF TEXT -- a logical unit from the data set which the user is working on with WYLBUR's help. The general form of this type of line is

<LINE NUMBER> <SINGLE BLANK> <CONTENT STRING> CR

The <LINE NUMBER> is an attribute of the line, used by WYLBUR to determine the position of the line relative to the rest of the lines in the data set. For example, all lines before a line with line number seven will have a line number less than seven and all lines following that line

E- 8

will be numbered greater than seven. A line number is a mixed decimal number with four integer digits and three decimal digits

0013.098 9999.999 0065.100

Leading zeros may be dropped so that

0013.098 and 13.098

are equivalent. Terminal zeros may also be dropped so that

0065.100 and 65.1

are equivalent. Embedded blanks are not equivalent to zeros so that

6 5. 1 and 605.01

are not equivalent. More comments will be made concerning blanks within line numbers when the <SINGLE SPACE> is discussed below. If the line number is an integer, the decimal point need not be typed so that

0098.000, 98.0, 98., and 98

are all equivalent.

The second major part of a line of text is the <CONTENT STRING> (or simply content) of the line. This may be a string of from 0 to 133 consecutive characters. If no characters are present, then the line is a line of blanks. It may be useful to the user to have WYLBUR monitor the number of characters which are contained in the content of a line. Consider, for example, the construction of a data set which is a Fortran program. Since a maximum of 72 characters may be in any line, it is useful to the user to

know when he has typed more than that number of characters in a line. For this reason, WYLBUR recognizes an attribute of lines called LENGTH - a count of the number of characters in the line, up to and including the last non-blank character. This attribute is assumed by WYLBUR to have a value of 72 unless the user takes definitive action (by using the SET command described below) to change it to some other value. When a line with more than the designated number of characters is typed, WYLBUR will warn the user that he has exceeded the limit which has been set. The line will still be accepted by WYLBUR and placed in its appropriate place in the data set -- the message that WYLBUR sends is a warning only. The user must change the line so that it conforms to the length which he desires. Note that LENGTH is only an attribute of the contents of a line - the characters used for the line number are not included in the length of a line.

The remaining part of a line of text is the <SINGLE BLANK>. This is used by WYLBUR to recognize the dividing point between the line number and the content of the line. Hence anything appearing before the blank is interpreted as a line number and anything after the blank is considered part of the content of the line. If three (say) blanks follow the line number, WYLBUR will use the first as a signal for the end of the line number and the next two blanks will cause the content of the line to begin with two blank characters. This is the reason that blanks are

unacceptable in a line number and that the line of text

1 0.1 THIS IS A DUMMY LINE

will not be line 100.1 with content

THIS IS A DUMMY LINE

but rather will be line 1.000 with content

0.1 THIS IS A DUMMY LINE

RANGE has already been defined as a group of lines which all possess some distinguishing characteristic. The most common instance of this is the EXPLICIT TYPE OF RANGE where the characteristic is that the line number lies within some range of numbers (and hence the name range). This type of range is denoted in WYLBUR by giving two explicit line numbers separated by the character /

1/10

In the example above, the lines of interest are those having a line number greater than or equal to 1.000 and less than or equal to 10.000. A special case is that in which there is only one line in the range

78.098/78.098

which may be abbreviated to just the single line number

78.098

In defining this type of range, the user may employ the special line numbers FIRST and LAST, which have as values the line numbers of the first and last lines in the data set on which the user is working. (F and L are legal abbreviations for FIRST and LAST.) This leads to the following possible ranges

FIRST/10
1.09/LAST
or FIRST/LAST

As a further convenience the user may also specify the range FIRST/LAST by using the special explicit range specification ALL. It should be emphasized that these special line numbers and the special range ALL are only legal within an explicit range specification.

A second form of an explicit range is just a list of line numbers.

F,10.2,75,114.125,126.01

WYLBUR requires that there be no more than ten numbers in such a list. Also, it is required that the line numbers are in ascending sequence.

The second type of range which may be specified is the ASSOCIATIVE RANGE in which the distinguishing characteristic is an instance of some <CHARACTER STRING>. This type of range is specified by enclosing the <CHARACTER STRING> within quote marks. A 2741 keyboard has two types of quote marks available - a single quote (') and a double quote ("). Either of these may be used to enclose the <CHARACTER STRING> but consistency must be maintained. Thus the following are valid associative ranges

"VARIABLE"
or 'VARIABLE'

whereas the following is not

"VARIABLE'

Also take note that blanks are significant in an

associative range specification, so that

" BLANKS ARE PART OF THE STRING "

and "BLANKS ARE PART OF THE STRING"

are not equivalent range specifications.

If a quote mark appears in the <CHARACTER STRING>, then the user must follow some additional rules. It is easiest to enclose the <CHARACTER STRING> in quote marks of the opposite type from those appearing in the <CHARACTER STRING>. If the user finds that he must use the same type of quote marks to enclose the string, then he should type those quote marks which appear in the <CHARACTER STRING> as two consecutive instances of the quote mark. For example

"DON'T"
and 'DON'T'

both designate that range of lines which contain the character string

DON'T

As an example of the meaning of various associative ranges consider the following text

1. DON'T WORRY IF YOU HAVEN'T UNDERSTOOD
2. EVERYTHING SO FAR. WYLBUR IS VERY UNDERSTANDING
3. AND WILL HELP YOU WHEN YOU MAKE A MISTAKE
4. DURING A SESSION BY GIVING YOU SOMEWHAT
5. INFORMATIVE ERROR MESSAGES INDICATING
6. WHAT WAS WRONG AND SUGGESTING A POSSIBLE METHOD
7. OF CORRECTING THE ERROR. TRYING TO
8. OPERATE WYLBUR WILL BE THE BEST WAY
9. YOU HAVE OF LEARNING HOW TO USE IT. BUT
10. DON'T THINK FROM THIS THAT YOU SHOULD
11. NOT READ THE REST OF THIS MANUAL BEFORE
12. ATTEMPTING TO USE WYLBUR.

The following associative range specifications give the following sets of lines

```

"DON'T"           <1,10>
'TH'              <2,6,7,8,10,11>
'TH'              <7,8,10,11>
"DON'T WORRY IF YOU HAVEN""T" <1>

```

The user may wish to restrict the lines included in an associative range to a certain subset of all lines which contain the <CHARACTER STRING>. He may do this by appending a series of digits which specify the ordinal number of each line which is to be in the subset. Thus, using the above example

```

'TH'              <2,6,7,8,10,11>
'TH' (1,3/5)     <2,7,8,10>
'TH' (1,3/6)     <2,7,8,10,11>

```

The user may further restrict the chosen lines by specifying an explicit range in which the search should be made

```

'TH' IN 7/10     <7,8,10>
'TH' (1,3) IN 7/10 <7,10>
'TH' IN ALL      <2,6,7,8,10,11>

```

It should be noted that if the user knows that the string of interest occurs in only a restricted section of his data set, he is doing WYLBUR a service to tell him this additional information. With this extra information, WYLBUR won't need the perhaps lengthy time required to search the full data set.

The user may further restrict an associative range by suffixing the string with column positions within which the string must be contained in any given line.

```

'TH' 6/8         <2,10>
'TH' 6/8 (1)    <2>
'TH' 6/8 (1) IN 7/10 <10>

```

Note that the string must begin and end within the specified column positions. If only one column position is specified, the string is required to start in that column position.

'TH' 6

<2>

Blank lines are considered by WYLBUR to be lines with null content. Thus the associative range '' or "" would retrieve all blank lines in the working data set. But the range ' ' or " " would retrieve all lines having one blank and at least one non-blank character, i.e. not blank lines.

The final hierarchical form which may be constructed is the DATA SET - the ultimate logical unit which WYLBUR can be used to construct. A data set is a group of lines, ordered in increasing sequence according to their line number attributes, and corresponds to some logical unit such as a Fortran or PL/I program or merely some data set of text. A data set has many attributes, the first of which is the name of the data set - the data set's DSNAME. The DSNAME can be up to 39 characters long and is of the form

A.B.Z

where A through Z must be character strings of less than eight characters and must begin with an alphabetic character.

The next attribute is the place of residence of the data set. This is called the VOLUME attribute of the data

set and denotes the alphameric name of some data set storage device such as a disk. Current issues of the Users' Bulletin give the VOLUME names which are available for use in storing private data sets.

The final attribute of a data set is the FORMAT in which it is written. This may be one of the following forms - EDIT, CARD, or PRINT. It should be emphasized that these formats only apply to the manner in which data sets are stored on external storage devices. When a user is working on one of his data sets with WYLBUR, he is actually working on a copy of the data set and this copy is held by WYLBUR in EDIT format.

For a data set to be in EDIT FORMAT means that the line number attributes are attached to the lines of text and that the content is stored in a special form.

CARD FORMAT consists (as the name implies) of card images - 80 character contents for the lines of text. If any line originally had a length greater than 80 characters, the 81st and greater characters will be lost. Line numbers are not stored with the line contents.

PRINT FORMAT is the same as CARD format, with the single exception that the length of the lines of text is 133 characters.

EDIT, CARD, and PRINT are the standard formats available in WYLBUR. In addition, the user may specify the exact number of characters in each record by using the LRECL type of format

LRECL=80 is equivalent to CARD format
LRECL=133 is equivalent to PRINT format
LRECL=122 would access records each having
122 characters

The normal WYLBUR user will ordinarily not have to concern himself with the use of LRECL format.

WYLBUR COMMANDS

There are three modes in which WYLBUR can be employed. The first is called COLLECT MODE and is that mode in which the user is having WYLBUR construct a new data set or add lines of text to an existing data set. The second is EDIT MODE in which the user is having WYLBUR make corrections to lines in an existing data set. This mode is the most important one of the three since it is that for which WYLBUR is primarily designed and intended. The third mode is APPLY MODE and is that in which the user asks, through WYLBUR, that some processing be done on the data set which the user has created. The following discussion of the WYLBUR commands is split into three sections corresponding to these three modes of WYLBUR operation.

In the following discussion, the data set upon which the user is making modifications is called the WORKING DATA SET. This is in contrast to EXTERNAL DATA SETS - all other data sets which exist in the system. External data sets are therefore those data sets external to the scope of attention of the user and his terminal.

SIGN ON PROCEDURE

When the user sits down at a terminal, he should first check that the COM-LOC switch on the side of the unit is in the COM position (switch up), enabling the terminal to communicate to the central processor. The other position of this switch, LOC (switch down), puts the terminal into local mode of operation in which it acts as a conventional electric typewriter. Next the user should check to see that the terminal is on by depressing the RETURN key or ATTN key. If nothing happens, this means that the ON-OFF toggle switch is in the OFF position and should be depressed to the ON position. When the terminal is on, depressing the RETURN key or ATTN key causes STP (the Stanford Terminal Processor - a systems program which acts as the interface between the user and the WYLBUR program) to start up the sign-on procedure, during which it asks the user a series of initial questions so that the validity of the user and his account number may be checked.

STP first types

STP NN MM/DD/YY HH:MM:SS

STP signifies that the Stanford Terminal Processor is processing the information as an interface between the user and WYLBUR. NN shows that the user is at a terminal which is attached to the central processor through a port having number NN. The date is given as month, day, and year - MM/DD/YY - and the current time is given as hour (24

hour clock), minute, and second - HH:MM:SS. STP follows this line with another giving the message of the day - special directions of which the user should be aware. Then STP requests that the user supply his name and the user should answer by typing his last name and a carriage return

NAME ? RIDDLE CR

STP asks the user to identify himself further by giving his account number

ACCOUNT ? T000 CR

The user is then asked to supply a keyword in order to determine that he has a right to use the account number which he just gave

KEYWORD ? F00 CR

(If the user has not set a keyword, then CR - a blank keyword - is the correct response.) As a protection feature, STP overprints both the account number and keyword with a series of miscellaneous characters. Unauthorized use of an account can therefore not be made by a person finding output which has been left in a public area. Provided that the correct response is made the user is asked to supply the number of the terminal which he is using

TERMINAL ? H15 CR

The response to this prompt is found on a label attached to the terminal at the right of the keyboard. The user's response to this prompt is used by STP to collect

statistics on the usage of various terminal units.

Associated with each valid account number is a list of ASSOCIATED TERMINAL NUMBERS. These are used for the purpose of insuring that each user can get his fair share of the computer's services. Looking at these lists in the reverse direction, there is a list of valid account numbers associated with each associated terminal number. These lists insure a uniformity of service since STP will allow only one user from each group assigned to an associated terminal number to be signed on at any one time. If a second user attempts to sign on he will receive the message

ASSOCIATED TERMINAL NUMBER ALREADY BEING USED

In this manner, one terminal which is rented for the use of many people, can not block the use of the system by other users who are renting other terminals.

If a user's account number has more than one associated terminal number, then, when he signs on, STP will consider him as using the first (the associated terminal numbers are stored in collating sequence) associated terminal number for his account number which is not already being used. If the user wishes to override this default decision he may explicitly specify an associated terminal number after his account number

ACCOUNT ? T000 H26

It should be reiterated that the user's response to the TERMINAL? prompt is used by STP to direct the collecting of

statistics on terminal usage and is not used to determine the associated terminal number which the user is considered to be using.

During the sign-on procedure, the user is conversing with STP. As a final step, STP asks the user which facility is to be used

SYSTEM? WYLBUR

Note that if only one system is available for use, STP does not ask the user this question.

This completes the sign-on procedure and STP now calls in WYLBUR as is indicated by the next prompt

COMMAND ?

by which WYLBUR signals that it is ready to accept directives from the user.

COLLECT MODE OF OPERATION

Now that the user has signed onto the WYLBUR system, let us suppose that he wishes to construct a new data set using WYLBUR. Since every session starts with a blank working data set, what the user wants to do is to input consecutive lines of text into this initially blank working data set. This mode of operation is what is called COLLECTING lines of text. The only thing which a user need do to enter COLLECT mode is give the COLLECT command

COMMAND ? COLLECT

WYLBUR will now prompt the <LINE NUMBER> and <SINGLE SPACE> parts of consecutive lines of text and expect that the user

will fill in the <CONTENT STRING> part for each line

1. ? THIS IS AN EXAMPLE
2. ? OF THE PROMPTS AND
3. ? ANSWERS WHICH TAKE PLACE
4. ? DURING THE COLLECT MODE
5. ? OF OPERATION.

As a convenience, the COLLECT command may be given in one of two shortened forms. The first (as with most of the WYLBUR commands -- watch for further examples below) is just the first three characters of the command word

COMMAND ? COL

The second is a command which is just an ATTN character

COMMAND ? ATTN***

Before considering other features of the collect mode of operation, it is appropriate to explain the function of the non-typing keys when they are used during the typing of lines of text.

Carriage return signals that the user has finished typing the contents into the line of text and wishes WYLBUR to prompt with the <LINE NUMBER> of the next line of text.

Back-spacing over a character erases it from the line of text which is being typed.

1. ? AN EXMPLE OF USE
AMPLE OF THE USE OF THE BACK-SPACE KEY

results in a line numbered 1.000 and having content

AN EXAMPLE OF THE USE OF THE BACK-SPACE KEY

since the user first typed out to the USE, then back-spaced to the beginning of MPLE and then typed the rest of the line. If the user back-spaces into the <LINE NUMBER> or

<SINGLE BLANK> part of the line of text (both of which were prompted by WYLBUR), he will lose all characters which he may subsequently type into these areas. The example

1.068?
98 ATTEMPT TO CHANGE THE LINE NUMBER

where the user has attempted to change the line number which WYLBUR prompted, will result in a line numbered 1.068 with content

ATTEMPT TO CHANGE THE LINE NUMBER

Likewise

1.068?
ATTEMPT TO OVERWRITE THE LINE NUMBER

results in a line numbered 1.068 with content

EMPT TO OVERWRITE THE LINE NUMBER

The attention key has two functions during COLLECT mode. If any characters are typed before hitting the ATTN, then the ATTN key signals to WYLBUR that the user wishes to erase the line which has just been partially typed

1. ? THIS SHOWS WHAT HAPPENS WHEN
2. ? THE ATTENTION KEY IS HIT BEFORE
3. ? COMPLETING A LINE WITH A
4. ? CARRIAGE RETURN ATTN***
4. ?

The ATTN at the end of line 4.000 causes WYLBUR to disregard what was typed into the line, and WYLBUR prompts again with the same line number. (It should be noted here that whenever characters are typed into a line -- command type as well as a line of text -- and the line is terminated with an ATTN, WYLBUR pays no attention to that line and returns the prompt that was just previously made.)

The second function of the ATTN key during COLLECT mode is to signal to WYLBUR that the user wishes to leave COLLECT mode. To give this signal, the user must type the ATTN key as the only character in the line. WYLBUR will then forget about the line numbered with the last prompt and ask the user for a new command

```

1. ? THIS SHOWS WHAT HAPPENS WHEN WYLBUR
2. ? IS SIGNALLED THAT THE USER WISHES
3. ? TO LEAVE THE COLLECT MODE.
4. ? ATTN***
COMMAND ?

```

At line 4.000 the user typed ATTN, WYLBUR then responded with the three asterisks, returned to a new line, and asked the user for his next directive. The data set will consist of three lines numbered 1.000, 2.000, and 3.000. A duality should be apparent -- an ATTN command is sufficient to get the user into COLLECT mode and an ATTN answer to a line number prompt while in COLLECT mode is sufficient to remove the user from COLLECT mode.

The above examples show that WYLBUR COLLECTs new lines of text into a blank working data set by beginning with a line numbered 1.000 and obtains line numbers for subsequent lines by adding 1.000. These values are under the control of the user and he may append modifying phrases to the COLLECT command in order to cause WYLBUR to use different values. Directing WYLBUR as follows

```
COMMAND ? COLLECT 2.098
```

will cause WYLBUR to collect lines of text with numbers 2.098, 3.098, 4.098, 5.098, etc. -- start at 2.098 and add

1.000 to get subsequent line numbers. Further modifying the command

COMMAND ? COLLECT 2.098 BY .001

causes the COLLECTing of lines of text numbered 2.098, 2.099, 3.000, 3.001, etc. -- start at 2.098 and add .001 to get subsequent line numbers. It should be noted that these modifying phrases may not be appended to the COLLECT command when the ATTN short form is used.

Suppose now that the user has constructed a data set (i.e. that his working data set is not empty), and that he wishes to add more lines of text. Suppose further that the working data set consists of seven lines of text numbered 1.1, 2.1, 3.1, 4.1, 5.1, 6.1, and 7.1. If the user issues a COLLECT directive, WYLBUR will prompt for lines to be added to the end of the working data set

COMMAND ? COLLECT
8.1 ? COLLECTING MORE LINES INTO THE DATA SET,
9.1 ? STARTING AT THE END OF THE CURRENT LINES.
10.1 ? ATTN***
COMMAND ?

The question arises of how WYLBUR derives the starting number of 8.1? The working data set has an attribute, LAST, which is defined to be the line number of the last line of text. 8.1 is therefore derived by taking this LAST attribute and adding the increment which is currently in effect (namely 1.000 unless the user specifies otherwise in the COLLECT command). As before, the user may change the increment

COMMAND ? COLLECT BY .1
7.2 ? COLLECTING MORE LINES, STILL INTO THE END
7.3 ? OF THE DATA SET, BUT NOW WITH A TEMPORARY
7.4 ? INCREMENT OF .1
7.5 ? ATTN***
COMMAND ?

The user may also collect lines into the interior of his working data set, but WYLBUR will not allow any existing lines to be overwritten or passed in the process. Consider the working data set as above and the command

COMMAND ? COLLECT 2.1 BY .1

Since collection of a line numbered 2.1 would cause an existing line to be overwritten, WYLBUR will return an error message and ask for a new command. If, on the otherhand, the command is

COMMAND ? COLLECT 2.2 BY .1

WYLBUR is quite happy to let the user add lines numbered 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, and 3.0; but will not collect a line numbered 3.1 since it would cause an existing line to be overwritten. Finally, if the command were

COMMAND ? COLLECT 2.2 BY .4

WYLBUR will collect lines numbered 2.2, 2.6, and 3.0, but will not pass the existing line 3.1 in order to collect a line numbered 3.4.

While COLLECTing lines the user may perhaps enter a line which contains more than LENGTH characters. Normally, WYLBUR will only issue a warning message, informing the user that this has occurred. However, the user may turn on

an OVERFLOW option (see the SET OVERFLOW command below) which will cause WYLBUR to place all the extra characters into the next line. Suppose that LENGTH is set at 40 and that OVERFLOW is set on.

```

COMMAND ? COLLECT 90 BY 10
 90.     ? THIS IS A LINE WHICH HAS 44 CHARACTERS IN IT
100.     ? IN IT AND ADD MORE
110.     ? ATTN***
COMMAND ? LIST 90/110
 90.     THIS IS A LINE WHICH HAS 44 CHARACTERS
100.     IN IT AND ADD MORE
COMMAND ?

```

Note first that breaks in the line containing excess characters are made only at blanks and never in the middle of a word. Also note that the overflowed characters are prompted by WYLBUR at the beginning of the next line. This overflow rectification process will work with any number of characters in the line - WYLBUR chooses successive groups of (approximately) LENGTH characters and makes them into new lines and then prompts for the next line, with the extra characters left over already present in the next line. If a string of more than LENGTH characters occurs without any blanks in it, WYLBUR will abort the overflow rectification process with an error message. The rectification done up to that point will be preserved, but the string itself and whatever follows it in the line just entered will be lost.

USING PREVIOUSLY DEFINED DATA SETS

Using WYLBUR would become rather tiresome if the only way in which a user could get a non-blank working data set

would be to type in the full text while in COLLECT mode. Therefore, it is possible to give a command to WYLBUR to have it call in a copy of an external data set to use as the working data set. This could be a data set which the user constructed at an earlier session, a data set which has been constructed by another user, or a card deck which has been read into the system (the procedure for accomplishing this last example will be given in an Appendix).

It should first be emphasized that only a copy of the external data set is made available to the user. If the user seriously damages the working data set by making disastrous changes (or if the system dies during the session - thereby destroying all working data sets), a version of the unimpaired data set still exists.

In order to obtain a copy of the external data set, the user should issue a USE command

COMMAND ? USE DUMMY

where DUMMY is the DSNAME of the data set which the user desires. The user's account number is appended to the front of the DSNAME to obtain the real name of the data set. In the above example, a data set named T000.DUMMY would be searched for on the external storage devices.

The user may help WYLBUR in its search for the data set by giving the VOLUME attribute

COMMAND ? USE DUMMY ON SYS07

WYLBUR will ask the user to supply the format of the data

set

```
COMMAND ? USE DUMMY ON SYS07
FORMAT ? EDIT
COMMAND ?
```

and after getting this information, WYLBUR will get a copy of the data set to use as the working data set for the user. The user may preempt WYLBUR's standard request for the data set format by stating it in the USE command.

```
COMMAND ? USE DUMMY EDIT
COMMAND ? USE DUMMY ON SYS07 EDIT
```

When WYLBUR obtains the copy, it is automatically put into EDIT format if it is not already in that format. This means that for CARD and PRINT format data sets, WYLBUR will attach line number attributes to every line of text. The line numbers supplied will begin with 1.000 and have an increment of 1.000 (the increment may be changed - see section below on the DELTA global parameter).

A user may obtain a copy of a public data set belonging to some other user by giving that data set's full DSNAME preceded by the & character

```
COMMAND ? USE &****.DUMMY
```

where **** must be replaced with the account number of the user who constructed the data set named DUMMY. In no cases will the original copy of the data set be scratched or overwritten by anyone but the user who constructed it.

The user may specify the LRECL type of format

```
COMMAND ? USE DUMMY LRECL=121 ON SYS05
```

But note that LRECL may not be greater than the number of

characters in the physical record or the record will be ignored. If the number of characters in the physical record is not an integral multiple of LRECL, those characters remaining after all possible lines of LRECL characters have been inserted into the working data set will be ignored.

When using data sets which are in PRINT, CARD, or LRECL format, the user may specify that a given number of records should be skipped when reading in the data set.

COMMAND ? USE DUMMY LRECL=80 SKIP=1000 ON SYS06

This will cause the first 1000 card images to be ignored and the card image, that without the SKIP phrase would have been line number 1001, will be line number 1.

GLOBAL PARAMETERS

Global parameters are adjustable features in WYLBUR which may be set to an appropriate value, and will remain set at that value for the duration of the session or until the user resets their value.

The first global parameter is the already explained LENGTH attribute of lines of text. The user may set this to any value between 1 and 133 (inclusive) by use of the SET command

COMMAND ? SET LENGTH = 80

If the user does not set the LENGTH attribute, it automatically has the value 72.

The second global parameter is the setting of the TABS

at the terminal. These are used as on a standard typewriter, but the user must follow a rather rigid procedure for telling WYLBUR what the settings are. This rigid procedure is necessitated by the way in which a 2741 works. When the TAB key is pressed at the terminal, the typing head is released and travels to the right. A protruding bar stops the movement of the typing head at the next tab setting. However, the only signal that is sent to the computer is one which indicates that the TAB key was pressed - no signal is sent to tell the computer where the typing head stopped. Thus the tabs may be physically set at the terminal, but WYLBUR will not know what the settings are until the user completes the following sequence. The user initiates this sequence by asking WYLBUR to set the tabs

COMMAND ? SET TABS

WYLBUR then gives the user directions and a line of column numbers

COMMAND ? SET TABS

TYPE A "1" BENEATH EACH POSITION AT WHICH YOU SET A TAB.

123456789A123456789B123456789C123456789D12...

START ?

The first thing the user must do is to clear the physical tab settings which already exist at the terminal. This is done by successively tabbing across the line and depressing the CLR toggle switch at each tab until the right-hand margin is reached. Then the user should hit ATTN, which will cause WYLBUR to return to the beginning of the next

line and re-prompt START?. The user then spaces over to each successive column in which he wants a tab set, first presses the SET toggle switch, and then types a "1"

```

COMMAND ? SET TABS
TYPE A "1" BENEATH EACH POSITION AT WHICH YOU SET A TAB.
-----
123456789A123456789B123456789C123456789D12...
START ?
START ?           1           1           1           1

```

The above example shows the successful setting of the tabs in columns 10, 20, 30, and 39.

Alternatively, the user may append to the SET TABS command a list of numbers representing the column numbers for the tab settings

```

COMMAND ? SET TABS = 10,20,30,39
COMMAND ?

```

But note that the physical tab settings at the terminal should correspond to the information given when using this form of the SET TABS command.

If the tabs have not been set and the tab key is used, WYLBUR will pay no attention to the line and will issue an error message. This occurs also if not enough tab stops have been set - for example, if only three tab stops have been set and the tab key is pressed four or more times.

The third global parameter is the increment which WYLBUR uses to derive one line number from its predecessor -- DELTA. The user may set DELTA

```

COMMAND ? SET DELTA = XXXX.XXX

```

where XXXX.XXX is any decimal number between .001 and 9999.999 (inclusive). If the user doesn't set DELTA, a

default value of 1.000 is used. So far, the only use of DELTA which has been encountered is in the COLLECT mode where it was noted that a DELTA of 1.000 was used unless the user overrode this value by specifying a different value in the COLLECT command (the BY phrase). In actuality, the DELTA used is the current value of the global parameter -- it was said to be 1.000 because that is the default value.

The OVERFLOW global parameter, used only in COLLECT mode, is turned on and off by means of a SET command

```
COMMAND ? SET OVERFLOW = ON
COMMAND ? SET OVERFLOW = OFF
```

OVER is a valid abbreviation for the word OVERFLOW.

The final global parameter is the LAST attribute of a working data set, which the reader should remember is the line number of the last line of text in the data set. The user may not set the value of this parameter explicitly -- whenever new lines are added to the end of a data set, the value of LAST is implicitly and automatically reset.

All examples of the use of WYLBUR are typed in upper case letters in this manual. This is the normal mode in which WYLBUR operates - all alphabetic characters, whether typed in upper or lower case are recognized by WYLBUR as being in upper case. If the user wishes to use both upper and lower case, he must tell WYLBUR

```
COMMAND ? SET UPLOW
```

Then when the user wishes to revert to normal mode, he

directs WYLBUR

COMMAND ? SET UPPER

Note that regardless of the case mode which is in effect, all typing in a command line is recognized by WYLBUR as upper case except those characters delimited by quote marks (as in associative ranges). Also note that, at present, the 1403 high-speed printers are only capable of printing upper case. Thus any text listed on the 1403 will be printed in upper case regardless of its actual case.

The discussion above indicates the manner in which the global parameters may be set by the user. How can the user find out the values of the parameters if he has forgotten them? This is done by asking WYLBUR to SHOW their values.

COMMAND ? COLLECT

- 1. ? THIS IS AN EXEMPLARY DATA SET USED
- 2. ? TO REVIEW THE SETTING AND SHOWING
- 3. ? OF GLOBAL PARAMETERS.
- 4. ? ATTN***

COMMAND ? SHOW TABS

CURRENT TAB SETTINGS ARE NONE.

COMMAND ? SHOW LAST

CURRENT LAST LINE NO. IS 3.

COMMAND ? SHOW LENGTH

CURRENT LENGTH IS 72.

COMMAND ? SHOW DELTA

CURRENT GLOBAL DELTA IS 1.

COMMAND ? SET DELTA = .01

COMMAND ? SHOW DELTA

CURRENT GLOBAL DELTA IS 0.01

COMMAND ? SET LENGTH = 80

COMMAND ? SHOW LENGTH

CURRENT LENGTH IS 80.

COMMAND ? COLLECT

3.01 ? EXTRA LINE ADDED

3.02 ? ATTN***

COMMAND ? SHOW LAST

CURRENT LAST LINE NO. IS 3.01

COMMAND ? SET TABS

TYPE A "1" BENEATH EACH POSITION AT WHICH YOU SET A TAB.

123456789A123456789B123456789C123456789D12...

```

START ?
START ?           1           1           1           1
COMMAND ? SHOW TABS
CURRENT TAB SETTINGS ARE - 10 - 20 - 30 - 39 -
COMMAND ? SHOW CASE
UPPER CASE ONLY
COMMAND ? SET UPLOW
COMMAND ? ATTN***
3.02 ? AB
3.03 ? ab
3.04 ? ATTN***
COMMAND ? LIST 'ab'
3.03 ab
COMMAND ? SHOW CASE
UPPER-LOWER CASE
COMMAND ?

```

EDITING OF DATA SETS

This is the heart of WYLBUR --- the primary reason that it was designed and implemented. It is by using these commands that the user may have the computer help him in changing invalid character strings and lines which his data set may contain. Since data sets will most likely be programs, WYLBUR is, in a sense, a glorified keypunch. But as the user reads the rest of this manual, it is hoped that he will see that using this appellation is a gross underestimation, since, when used to its full capabilities, WYLBUR is capable of much more than mere string replacement.

WYLBUR's editing commands are of two distinct types, differentiated by whether the operation called for is effected upon full lines of text or merely upon strings which may appear in the content of a line of text -- INTER-LINE and INTRA-LINE editing.

INTER-LINE EDITING

The most common thing that the user will probably want to do with his data set is to list it. This he can do by commanding WYLBUR to LIST

COMMAND ? LIST

1. THIS IS A DUMMY DATA SET WHICH SHOWS
2. WHAT HAPPENS IN RESPONSE TO A LIST
3. COMMAND. EACH LINE OF TEXT IS PRINTED
3.1 P
4. WITH ITS ASSOCIATED LINE NUMBER UNTIL
5. ALL THE LINES IN THE DATA SET ARE
6. LISTED, AT WHICH POINT WYLBUR PROMPTS
7. THE USER FOR ANOTHER DIRECTIVE.

COMMAND ?

The user may desire that only an explicit range within the data set be listed

COMMAND ? LIST 3/5

3. COMMAND. EACH LINE OF TEXT IS PRINTED
3.1 P
4. WITH ITS ASSOCIATED LINE NUMBER UNTIL
5. ALL THE LINES IN THE DATA SET ARE

COMMAND ?

The user may further wish that his listing not contain the line numbers of the lines of text

COMMAND ? LIST UNNUMBERED

THIS IS A DUMMY DATA SET WHICH SHOWS
WHAT HAPPENS IN RESPONSE TO A LIST
COMMAND. EACH LINE OF TEXT IS PRINTED
P
WITH ITS ASSOCIATED LINE NUMBER UNTIL
ALL THE LINES IN THE DATA SET ARE
LISTED, AT WHICH POINT WYLBUR PROMPTS
THE USER FOR ANOTHER DIRECTIVE.

COMMAND ? LIST 3/5 UNNUMBERED

COMMAND. EACH LINE OF TEXT IS PRINTED
P
WITH ITS ASSOCIATED LINE NUMBER UNTIL
ALL THE LINES IN THE DATA SET ARE
COMMAND ?

Or as an alternative (albeit rather useless in this

context) the user may wish to have just the line numbers listed

COMMAND ? LIST NOTEXT

1.

2.

3.

3.1

4.

5.

6.

7.

COMMAND ? LIST 3/5 NOTEXT

3.

3.1

4.

5.

COMMAND ?

This NOTEXT option is more useful when the user wants to find the line numbers of all lines which contain a certain character string - i.e. those lines in an associative range - a case which is discussed next.

Interesting uses of WYLBUR's listing capabilities arise when associative ranges are used

COMMAND ? LIST 'DATA SET' IN ALL

1. THIS IS A DUMMY DATA SET WHICH SHOWS

5. ALL THE LINES IN THE DATA SET ARE

COMMAND ?

The above example shows that WYLBUR can be used to find instances of some character string which is of interest to the user. WYLBUR has found all occurrences in the specified range (in this case all) and printed their image for inspection. The user may also restrict the range in which WYLBUR is to search

COMMAND ? LIST 'DATA SET' IN 4/7

5. ALL THE LINES IN THE DATA SET ARE

COMMAND ?

The user could also ask for only the line numbers of the lines containing the string

```
COMMAND ? LIST 'DATA SET' IN ALL NOTEXT
1.
5.
COMMAND ?
```

Likewise, the user could append the phase UNNUMBERED instead of NOTEXT and receive the line contents only.

When listing with an UNNUMBERED option, the user may specify a MARKER, an instance of which in the first column of any line in the range will cause the listing to be temporarily suspended. After suspension, the listing may be resumed with a CR command.

```
COMMAND ? LIST ALL MARKER = P UNNUMBERED
THIS IS A DUMMY DATA SET WHICH SHOWS
WHAT HAPPENS IN RESPONSE TO A LIST
COMMAND. EACH LINE OF TEXT IS PRINTED
CR
WITH ITS ASSOCIATED LINE NUMBER UNTIL
ALL THE LINES IN THE DATA SET ARE
LISTED. AT WHICH POINT WYLBUR PROMPTS
THE USER FOR ANOTHER DIRECTIVE.
COMMAND ?
```

Note that the line containing the MARKER in column one is not printed and that the MARKER character is only recognized when it occurs in column one.

After a little use of WYLBUR, it should be apparent to the user that having it list all of a very large data set is not very convenient due to the 15 character per second speed of the 2741. Therefore the user may command

```
COMMAND ? LIST OFFLINE
TYPE YOUR BIN NUMBER FOR OFFLINE LIST.
BIN NO. ? 333
XXX IS THE JOB NUMBER FOR YOUR LIST OFFLINE.
COMMAND ?
```

This causes WYLBUR to append the necessary JCL cards to the data set and put the job into the job queue for HASP. The output will be marked as belonging in the bin which the user has specified, and the user may monitor the progress of his job by using the LOCATE command (see below). The user may preempt WYLBUR's standard request for a bin number by supplying it in the command prefaced by BIN or B

COMMAND ? LIST OFFLINE BIN 333

Normally the user will want the computer time required for the offline list to be charged to the same account number that was used in the sign on procedure. He may specify otherwise by giving an account number in the command prefaced by ACCOUNT, ACCT, or A

COMMAND ? LIST OFFLINE BIN 333 ACCT T000

If the user doesn't supply this ACCOUNT phrase, WYLBUR assumes that the account number given in the sign on procedure is the correct one to use.

When listing offline, the user may position his output on the printer paper by specifying a count of the number of blanks to be inserted at the left.

COMMAND ? LIST OFFLINE (5) B 333

will produce output which is shifted 5 positions to the right. The number of blanks specified must lie in the range 1 to 70. In addition, if an unnumbered offline list is done, a zero blank count may be specified thus allowing the user to list a file which contains its own printer

control characters. (When a LIST OFFLINE UNNUMBERED (0) is done with a data set that has been created by any of the IBM assemblers, WYLBUR needs to be told with an MC phrase

COMMAND ? LIST OFFLINE UNN (0) MC

since the carriage control characters output by the assemblers are not the standard ones used by the 1403 highspeed printers. WYLBUR passes this phrase on to the printers so that they can correctly interpret the carriage control characters.)

In a LIST OFFLINE command, the user may specify a title which will be printed at the top of the first page of the output.

COMMAND ? LIST OFFLINE B 333 'DUMMY - AUG 8, 1967'

Note that the TITLE option is only valid in an OFFLINE list, and that there may not be more than 60 characters in the TITLE.

The user may obtain a punched card deck of his working data set by giving WYLBUR a PUNCH command

COMMAND ? PUNCH 2/L UNNUMBERED B 333 ACCT T000

If the UNNUMBERED option is not specified, the line numbers will be punched into columns 73 - 80, overwriting whatever was in those columns. The BIN and ACCOUNT options work exactly as in the LIST OFFLINE command. The PUNCH command will automatically produce an offline listing of the cards punched.

The processing called for in both the LIST OFFLINE and PUNCH commands is twofold. First a short job is run to

produce the print and/or punch file. Then the created file is printed and/or punched. The user may attach a priority - URGENT, PRIORITY, EXPRESS (only for the execution phase), STANDARD, IDLE, or OVERNITE - to each of these stages. STANDARD is the default priority for jobs entered by the card readers, EXPRESS is the default priority for jobs entered via WYLBUR, and the others are successively higher and lower. Note that EXPRESS is only valid when the execution time for the job does not exceed two minutes, otherwise the priority will be changed to STANDARD. (A fuller explanation of these priority levels and their concomitant charges is in the main text of this User's Manual.) The user specifies these priorities by attaching two phrases to the command.

COMMAND ? LIST OFFLINE P B 333 U

Note first that the initial letter of each priority option is a legal abbreviation (except for EXPRESS which is abbreviated X). The above command calls for PRIORITY processing and URGENT printing - printing as soon as a 1403 is available.

The print stage of the processing may be deleted altogether (note that it's senseless to do this in a LIST OFFLINE command) by giving a print priority of NOPRINT.

COMMAND ? PUNCH B 333 O N

This would cause the punch job to be run overnight and the offline listing of the cards punched to be deleted.

If both the run and print priorities are to be the

same, the user need only specify it once. This single priority will be taken to apply to both stages. If the NOPRINT priority is given as the only priority phrase, this will cause the run priority to be EXPRESS and the print priority to be NOPRINT.

The user may request WYLBUR to change the line numbers of the lines of text in his working data set by giving the NUMBER command

COMMAND ? NUMBER

This causes the renumbering of the data set, starting at 1.000 and using the current value of the DELTA parameter. The user may specify another number at which the numbering is to begin by giving it in the command

COMMAND ? NUMBER 2.065

which will cause WYLBUR to start at a value of 2.065 for the first new line number. In addition, the user may temporarily override the setting of the global parameter DELTA and ask WYLBUR to use another value

COMMAND ? NUMBER 2.065 BY .01

It should be emphasized that this command causes the renumbering of the total data set and that the user may not specify that only a range (either explicit or associative) be renumbered.

The user may delete lines in his data set by giving the DELETE command

COMMAND ? DELETE 5.03/6.08

(Associative ranges may also be specified in the DELETE

command.) This will cause WYLBUR to remove from existence all the lines which lie in the specified range.

Opposite to the DELETE command is the INSERT command

```
COMMAND ? INSERT 13.02
13.02 ?
```

The user has specified that he wishes to insert a line numbered 13.02 and WYLBUR responds with that number and expects the user to fill in the contents of the line as would be done in COLLECT mode. WYLBUR will not allow the user to insert a line which has the same number as an existing line in the data set. Note that only a single line may be inserted -- the user may not specify that an explicit range of lines be inserted. (Remember that this can be done by COLLECTing the lines into the data set.) If the user attempts to insert a line with the same number as an existing line, WYLBUR prompts with an error message telling the user that he should use the REPLACE command (to be described next).

Frequently, the user will want to delete the contents of a line and then put another line into the data set with the same number but with new contents -- i.e. replace the contents of some line. This he may do in a single step by issuing the REPLACE command

```
COMMAND ? REPLACE 13.06
13.06 ?
```

which causes WYLBUR to overwrite the old contents of the line specified with the new contents which the user types

in response to the line number prompt. The user may specify that a range of lines be replaced, in which case WYLBUR will prompt successive line numbers from the specified range until the range is exhausted. As an example, suppose the data set has lines numbered 1.02, 1.03, 1.031, 1.032, 1.033, 1.04, and 1.05. Then commanding

COMMAND ? REPLACE 1.02/1.03

will result in WYLBUR requesting new contents for lines 1.02 and 1.03. On the otherhand, requesting

COMMAND ? REPLACE 1.03/1.05

will cause WYLBUR to prompt for new contents of lines 1.03, 1.031, 1.032, 1.033, 1.04, and 1.05. The user is not allowed to replace the contents of a line which does not exist in the data set - i.e. use the REPLACE command to insert. Rather, he should use the INSERT command.

If the user types just ATTN as the new contents of the line, the old contents of the line are not replaced -- the REPLACE command is aborted.

There are short forms for the DELETE, INSERT and REPLACE commands. The first is by giving only the first three characters of the command word. The second form is the same for all commands and consists of giving the full line of text for the inserted or replaced line as a response to the command prompt from WYLBUR

COMMAND ? 13.06 PUT IN THIS LINE

This would cause WYLBUR to put this new line image into the

data set in its appropriate place which is determined by the line number. If 13.06 already exists in the data set, then the old line will be replaced; and if 13.06 doesn't already exist, the line will be inserted into the data set. The user should be warned of two possible sources of error in using this command.

1. Be sure to terminate the line number with a single blank. Anything before the first blank will be used as the line number and anything following it will become the contents of the new line.

2. Don't use tabs in typing in the line contents, since the typing of the line number has displaced the characters in the content string by as many spaces as are used for the line number. Tabs will still work, but the user will find the contents displaced to the left of where he desires them.

If the user follows the line number with a single blank and then a carriage return, the specified line will be deleted. (beware !!!)

The user may find it necessary to copy the contents of a range of lines into another section of his working data set and may ask WYLBUR to do this for him. Consider the data set as

```

1.01 THIS IS A DUMMY DATA SET WHICH WILL BE USED
1.02 TO SHOW THE EFFECT OF VARIOUS COPY COMMANDS.
2.0  RAND
2.01 SAMPLE
2.02 RUNIN
3.0  SRFREPP
6.0  DATA SET

```

Listed below are various COPY commands followed by a listing of the resultant data set. It is assumed that each COPY command is given when the data set is in the form

given above.

COMMAND ? COPY 2.0/2.02 TO 47

COMMAND ? LIST

1.01 THIS IS A DUMMY DATA SET WHICH WILL BE USED
1.02 TO SHOW THE EFFECT OF VARIOUS COPY COMMANDS.
2.0 RAND
2.01 SAMPLE
2.02 RUNIN
3.0 SRFREPP
6.0 DATA SET
47.0 RAND
48.0 SAMPLE
49.0 RUNIN

COMMAND ? COPY 3/7 TO 1.03 BY .001

COMMAND ? LIST

1.01 THIS IS A DUMMY DATA SET WHICH WILL BE USED
1.02 TO SHOW THE EFFECT OF VARIOUS COPY COMMANDS.
1.03 SRFREPP
1.031 DATA SET
2.0 RAND
2.01 SAMPLE
2.02 RUNIN
3.0 SRFREPP
6.0 DATA SET

COMMAND ? COPY 2.02 TO 4.01

COMMAND ? LIST

1.01 THIS IS A DUMMY DATA SET WHICH WILL BE USED
1.02 TO SHOW THE EFFECT OF VARIOUS COPY COMMANDS.
2.0 RAND
2.01 SAMPLE
2.02 RUNIN
3.0 SRFREPP
4.01 RUNIN
6.0 DATA SET

COMMAND ? COPY 'DATA SET' IN ALL TO 2.03 BY .001

COMMAND ? LIST

1.01 THIS IS A DUMMY DATA SET WHICH WILL BE USED
1.02 TO SHOW THE EFFECT OF VARIOUS COPY COMMANDS.
2.0 RAND
2.01 SAMPLE
2.02 RUNIN
2.03 THIS IS A DUMMY DATA SET WHICH WILL BE USED
2.031 DATA SET
3.0 SRFREPP
6.0 DATA SET

(This example is actually wrong. The reader should try it on WYLBUR and figure out why the correct result occurs. If he can't figure out

the reason for the result, he should ask one of the SCC consultants.)

```

COMMAND ? COPY 'DATA SET' IN 1/5 TO 33 BY .001
COMMAND ? LIST
-----
1.01 THIS IS A DUMMY DATA SET WHICH WILL BE USED
1.02 TO SHOW THE EFFECT OF VARIOUS COPY COMMANDS.
2.0  RAND
2.01  SAMPLE
2.02  RUNIN
3.0   SREREPP
6.0   DATA SET
33.0  THIS IS A DUMMY DATA SET WHICH WILL BE USED

```

The user may copy lines from an external data set by giving a FROM phrase which specifies the external data set

```

COMMAND ? COPY 1/56 TO 35 BY .1 FROM DUMMY

```

This would result in all lines which are numbered between 1.000 and 56.000 in the external data set named DUMMY to be copied into the working data set starting at line 35 and occupying succeeding lines whose line numbers are automatically determined with a DELTA of .1. As with the COLLECT mode, the user is prevented from overwriting or interleaving existing lines in the working data set. Another rule concerning the COPY command is that any external data set which is specified must be in EDIT format. WYLBUR will return an error message if it is not, and the user must then get the external data set into EDIT format in order to reissue the command. This may be accomplished by using the data set (after saving the current working data set), which will cause WYLBUR to convert the data set to EDIT format, and then saving this edited version of the external data set under a new name in

EDIT format.

If the user wishes to copy a range of lines and then delete the old copy of the lines, he may do this with a single command - MOVE. This functions exactly as the COPY command with the following exceptions.

1. The old lines which were copied are deleted.
2. It is not possible to apply the MOVE command to an external data set, since a basic axiom is that the user may take destructive action only upon the current working data set.

Following are some examples of the MOVE command as applied to the exemplary data set used in the explanation of the COPY command

```

COMMAND ? MOVE 3/7 TO 1.03 BY .001
COMMAND ? LIST
1.01 THIS IS A DUMMY DATA SET WHICH WILL BE USED
1.02 TO SHOW THE EFFECT OF VARIOUS COPY COMMANDS.
1.03 SRFREPP
1.031 DATA SET
2.0 RAND
2.01 SAMPLE
2.02 RUNIN
COMMAND ? MOVE 'DATA SET' IN 1/5 TO 33 BY .001
COMMAND ? LIST
1.02 TO SHOW THE EFFECT OF VARIOUS COPY COMMANDS.
1.03 SRFREPP
2.0 RAND
2.01 SAMPLE
2.02 RUNIN
33. THIS IS A DUMMY DATA SET WHICH WILL BE USED
33.001 DATA SET

```

The TO phrase of both the COPY and MOVE commands may specify the special line number END. END is defined as DELTA higher than the current LAST of the working data set.

INTRA-LINE EDITING

This completes the discussion of the inter-line editing

commands available in WYLBUR. This group is powerful, by itself, as a tool to help the user edit his data sets, but requires that the full contents of the corrected line be typed when making changes. Helping the user make changes such as this in a more convenient manner is the function of the intra-line editing commands.

The first means of modifying the contents of a line is by use of the MODIFY command

```
COMMAND ? MODIFY 23.54
 23.54 THIS LINE IS TOOO BE MODIFID
ALTERS ?
```

which causes WYLBUR to print out an image of the line as it stands, and then prompt for alterations from the user. The first thing that the user wants to do to the above line is to remove the extra 0 characters. This is called deleting characters from the content and is accomplished by typing a D underneath the characters to be removed

```
COMMAND ? MODIFY 23.54
 23.54 THIS LINE IS TOOO BE MODIFID
ALTERS ? DD
 23.54 THIS LINE IS TO BE MODIFID
ALTERS ?
```

Actually the user need type a D only under the first and last character of the string which is to be deleted -- everything between the two D's will also be deleted

```
COMMAND ? MODIFY 23.54
 23.54 THIS LINE IS TOOO BE MODIFID
ALTERS ? DD
 23.54 THIS LINE IS TO BE MODIFID
ALTERS ? D D
 23.54 THIS LINE IS TO BE MOD
ALTERS ?
```

The user may also insert new characters into the line by

first typing an I underneath the character before which the insertion is to be made and following this with the character string to be inserted

```

COMMAND ? MODIFY 23.54
 23.54 THIS LINE IS TOOO BE MODIFID
ALTERS ?           DD
 23.54 THIS LINE IS TO BE MODIFID
ALTERS ?           D  D
 23.54 THIS LINE IS TO BE MOD
ALTERS ?           I NOW
 23.54 THIS LINE IS NOW TO BE MOD
ALTERS ?

```

Finally the user may replace a string in the line contents by typing an R under the first character to be replaced and following this with a string of characters which are the replacements. As many characters are replaced, starting with the one over the R, as there are characters in the string

```

COMMAND ? MODIFY 23.54
 23.54 THIS LINE IS TOOO BE MODIFID
ALTERS ?           DD
 23.54 THIS LINE IS TO BE MODIFID
ALTERS ?           D  D
 23.54 THIS LINE IS TO BE MOD
ALTERS ?           I NOW
 23.54 THIS LINE IS NOW TO BE MOD
ALTERS ?           RBEING MODIFIED
 23.54 THIS LINE IS NOW BEING MODIFIED
ALTERS ?

```

Using the R indicator restricts the user to replacing a number of characters with the same number of characters. If the user wants to do a replacement with a different number of characters he may use a combination of the D and the I indicators

```

COMMAND ? MODIFY 23.54
 23.54 THIS LINE IS TOOO BE MODIFID
ALTERS ?           DD

```

```

23.54 THIS LINE IS TO BE MODIFID
ALTERS ? D D
23.54 THIS LINE IS TO BE MOD
ALTERS ? I NOW
23.54 THIS LINE IS NOW TO BE MOD
ALTERS ? RBEING MODIFIED
23.54 THIS LINE IS NOW BEING MODIFIED
ALTERS ? D DIIN THE PROCESS OF
23.54 THIS LINE IS IN THE PROCESS OF BEING MODIFIED
ALTERS ?

```

The D D caused the word NOW to be deleted and I under the blank following the word NOW caused the string following the I to be inserted before the blank. The user need not type the second D to indicate the range of the deletion since the occurrence of the I indicator will automatically cause the deletion range to be terminated in the character position just before the I occurs. The user may also follow the D range with a replacement indicator and a string of replacement characters

```

COMMAND ? MODIFY 23.54
23.54 THIS LINE IS TOOO BE MODIFID
ALTERS ? DD
23.54 THIS LINE IS TO BE MODIFID
ALTERS ? D D
23.54 THIS LINE IS TO BE MOD
ALTERS ? I NOW
23.54 THIS LINE IS NOW TO BE MOD
ALTERS ? RBEING MODIFIED
23.54 THIS LINE IS NOW BEING MODIFIED
ALTERS ? D DIIN THE PROCESS OF
23.54 THIS LINE IS IN THE PROCESS OF BEING MODIFIED
ALTERS ? D RHAS BEEN
23.54 THIS LINE HAS BEEN MODIFIED
ALTERS ?

```

The user signals that he has made all alterations to the line by not typing anything into the ALTERS line but giving only a carriage return

```

ALTERS ? CR
COMMAND ?

```

The above example shows that WYLBUR always types out the modified image of the line after the user has specified the alteration. This may be suspended by typing an N before giving any alteration indicators

```

COMMAND ? MODIFY 23.54
23.54 THIS LINE IS TOOO BE MODIFID
ALTERS ?          N          DD
ALTERS ?

```

The position of the N is of no concern to WYLBUR as long as it appears before any other characters on the alteration line. On the other hand, the user may get a copy of the modified line, as it stands after correction, by typing a few blanks into the alteration line before hitting RETURN (remember that just hitting RETURN causes the modification process to be terminated)

```

COMMAND ? MODIFY 23.54
23.54 THIS LINE IS TOOO BE MODIFID
ALTERS ?          N          DD
ALTERS ?          CR
23.54 THIS LINE IS TO BE MODIFID
ALTERS ?

```

The ATTN key is also used during the modification process to signal directions to WYLBUR. First, if it is preceded by any other characters in the alteration line, WYLBUR does not pay any attention to that line (remember the discussion of the similar use of the ATTN key in COLLECT mode)

```

COMMAND ? MODIFY 23.54
23.54 THIS LINE IS TOOO BE MODIFID
ALTERS ?          DDATTN***
ALTERS ?          DD
23.54 THIS LINE IS TO BE MODIFID
ALTERS ?

```

If on the otherhand, no characters are typed into the alteration line before the ATTN key is hit, WYLBUR will erase all modifications which have been made to the line and suspend the modification process

```

COMMAND ? MODIFY 0.09
  0.09  RENEGE ON MODIFICATIONS ALREADY MADE
ALTERS ?           D       D
  0.09  RENEGE ON MODS ALREADY MADE
ALTERS ? ATTN***
COMMAND ? LIST 0.09
  0.09  RENEGE ON MODIFICATIONS ALREADY MADE
COMMAND ?

```

The user should note that he may make only one alteration per ALTERS prompt with the exception of the two examples previously given -- a deletion indicator followed by either an insertion or replacement indicator.

The user may also modify a range of lines (either associative or explicit) and WYLBUR will operate in the same manner as in the replacement of a range of lines -- it will prompt successive lines in the range until the range is exhausted.

Often the user will wish to modify every occurrence of some string in some area of his data set or perhaps in his entire data set (consider as an example the changing of an implicit integer variable name to an implicit floating point variable name in a Fortran program). He may use the MODIFY command to do this by specifying an associative range and then making the appropriate alterations when each line is brought out for inspection by WYLBUR. Thus if the string to be changed is MAX and it is to be changed to

FMAX, the user could direct

COMMAND ? MODIFY 'MAX' IN ALL

and make the changes to each line found and prompted by WYLBUR. He may do the same function in a much easier manner by issuing a CHANGE command to WYLBUR

COMMAND ? CHANGE 'MAX' TO 'FMAX' IN ALL

which will cause WYLBUR to search the specified range (in this case, the entire data set) and change every occurrence of MAX in every line in the range to FMAX. WYLBUR will also print the new image of every line which it changes during the process

COMMAND ? CHANGE 'MAX' TO 'FMAX' IN ALL

1.36 FMAX = 0

10.01 FMAX = FMAX(A(1), B(1), FMAX)

999. FMAX = FMAX/NSAMP

COMMAND ?

If the user does not wish to have the new images printed out for the changed lines, he may append a NOLIST option to the CHANGE command

COMMAND ? CHANGE 'MAX' TO 'FMAX' IN ALL NOLIST

COMMAND ?

The user may wish to restrict the changing of the occurrences to only a section of the data set which he may do by giving a range specification rather than the phrase ALL

COMMAND ? CHANGE 'MAX' TO 'FMAX' IN 999/9999

999. FMAX = FMAX/NSAMP

COMMAND ?

The user may further wish to change only a specific occurrence of the string in each line. This is done by

modifying the string to be changed with a number giving the ordinal position of the occurrence which is to be changed. Consider the result derived in the next to last example and the case that the user wishes to change the function name back to MAX.

```
COMMAND ? CHANGE 'FMAX' (2) TO 'MAX' IN 10.01  
10.01 FMAX = MAX(A(1),B(1),FMAX)  
COMMAND ?
```

The (2) specifies that only the second occurrence of FMAX is to be changed.

The user may be confused by an ambiguity between ordinal positions of character strings to be changed and the ordinal position of the subset of lines in an associative range. Consider the following possible command

```
COMMAND ? CHANGE 'MAX' (2) TO 'FMAX' IN 'MAX' (2/6)
```

What does it mean? First of all WYLBUR is told to direct its attention to the second through sixth lines in the data set which contain an instance of the string MAX. Within this subset of lines, WYLBUR is told to change the second occurrence of the string MAX in each line to FMAX.

Frequently, a user may desire to change the characters in a certain group of character positions within a range of lines. This may be done by specifying the numbers of the first and last character positions in the group to be affected

```
COMMAND ? CH 2/4 TO 'MX' IN 10/999  
10.01 ? FMX = FMAX(A(1),B(1),FMAX)  
999 ? FMX = FMAX/NSAMP  
COMMAND ?
```

If 2/2 had been specified as the group of character positions, only column 2 would have been changed. Further, if only 2 had been specified, the string MX would have been inserted before column 2 in the old line

```
COMMAND ? CH 2 TO 'MX' IN 10/999
10.01 ? FMXMAX = FMAX(A(1),B(1),FMAX)
999. ? FMXMAX = FMAX/NSAMP
COMMAND ?
```

The user may combine the string and column position notations to restrict the change to a string which is contained in a specified group of columns. Using the original example data set given above, consider the following exemplary commands

```
COMMAND ? CH 'MAX' 7/10 TO 'FMAX' IN ALL
1.36 FMAX = 0
10.01 FMAX = MAX(A(1),B(1),MAX)
999. FMAX = MAX/NSAMP
COMMAND ? CH 'MAX' 11/72 (2) TO 'FMAX' IN 10.01
10.01 FMAX = MAX(A(1),B(1),FMAX)
COMMAND ?
```

In the first change, every occurrence of MAX in columns 7 through 10 (note that the string must be wholly contained within the specified group of character positions) is changed. In the second, only the second occurrence of MAX which occurs in columns 11 through 72 is changed. The first change could be effected by commanding

```
COMMAND ? CH 'MAX' 7 TO 'FMAX' IN ALL
```

since, if only one column position is specified, the string 'MAX' is restricted to begin in that column position.

APPLY COMMANDS

These commands, as the name implies, are used to

request extra processing on the user's working data set. This is directly analogous to the situation in which the user has extra processing done on a card deck by submitting it to the computer through dispatch or the card readers.

The most important such command is the RUN command which directs WYLBUR to put the user's data set into the input stream of the HASP monitor

COMMAND ? RUN

This is exactly the same as submitting a program through dispatch or the card readers. The data set must be a logically complete unit containing all of the necessary JCL cards just as would be necessary in submitting a run. WYLBUR merely puts the data set into the input stream and informs the user of the job number which has been associated with the data set so that the user may monitor the job's progress.

COMMAND ? RUN

256 IS THE JOB NUMBER FOR YOUR RUN.

COMMAND ?

When the RUN command is given, WYLBUR automatically makes a copy (in CARD format) of the working data set so that the data set is acceptable to HASP. (The user should be careful not to run a data set which contains lower case characters since they are not converted to upper case.) In addition, the line numbers are automatically put into character positions 73 through 80, overwriting whatever was in these positions. Thus the output from the run will

contain the line numbers to facilitate further editing. If the user does not wish the line numbers to overwrite character positions 73 through 80, he must append an UNNUMBERED phrase to the command

COMMAND ? RUN UNNUMBERED

Run and print priorities may be attached to the RUN command in the same way that they are attached to the PUNCH and LIST OFFLINE commands

COMMAND ? RUN URGENT NOPRINT

After the job has been submitted through WYLBUR, the user may wish to find out what (if any) processing has been done on the job. Thus he may ask WYLBUR

COMMAND ? LOCATE 256

and WYLBUR will respond with the current status of the job -- it is waiting, it is finished execution, it is waiting for the printer, it is on the printed, etc. If WYLBUR can not find the job requested, it will tell the user that the job has most likely finished all its associated processing.

In the LOCATE command, the user may also search for a job by its job name rather than its job number

COMMAND ? LOCATE RUN1

JOB 118

JOB 214

JOB 226 IS AWAITING EXEC

This causes the batch stream to be searched for all jobs which have the name specified by the user. The job number of each is printed and the status of the last one found is determined and printed for the user.

Before running a job, the user may wish to send a message to the operators concerning tape mounts, etc. This is done with a variant of the SHOW command

COMMAND ? SHOW OPERATOR 'JOB MANUAL WILL NEED TAPE 231'

In general, this message sending capability should be used to communicate the special instructions which the user would put on the job request card of a batch job.

The user may wish to determine what load exists in the system. This is done by asking WYLBUR to show the status of the system.

COMMAND ? SHOW STATUS
JOB 159 IS BEING PRINTED
JOB 181 IS BEING EXECUTED
JOB 1 IS BEING EXECUTED
112 JOB(S) AWAITING EXEC, 4 HRS 29 MIN ESTIMATED
1 JOB(S) AWAITING PRINT, 25747 LINES
40 PERCENT SPOOL DISK(S) UTILIZATION
COMMAND ?

This command, coupled with the LOCATE command, gives the user a means of finding his job and determining a reasonable estimate of when it will be executed.

The user may determine the queues waiting for execution and printing, broken down into priority classes, by issuing the appropriate SHOW commands

COMMAND ? SHOW RUN
URGENT 3
PRIORITY 4
EXPRESS 20
STANDARD 35
IDLE 20
OVERNITE 25
COMMAND ? SHOW PRINT
URGENT 3
PRIORITY 6
STANDARD 20
IDLE 35

OVERNITE 20
COMMAND ?

This information will help the user in deciding what priorities to attach to his program.

The second apply mode command is ALIGN. This command causes WYLBUR to process an explicit range in the working data set so that at most LENGTH characters are on each line. Splits in a line are always made at a blank - a word is never split between two lines. In addition, any line which has a special MARKER character in its first column (this MARKER character must be specified in the ALIGN command if this option is to be used) is not aligned and will cause the alignment process to be restarted at that line. In addition to the special MARKER character specified by the user, a blank in column one (as in indented material) causes the alignment process to be restarted.

COMMAND ? SET LENGTH = 80
COMMAND ? CLEAR
COMMAND ? COL 11
11. ? P THIS IS A SAMPLE DATA SET USED TO
12. ? SHOW THE EFFECTS OF
13. ? POSING THE ALIGN
14. ? COMMAND TO WYLBUR.
15. ? P THE P MARKERS IN
16. ? COLUMN ONE CAUSE THE RESTARTING
17. ? OF THE
18. ? ALIGNMENT PROCESS. BUT
19. ? NOTE THAT ONE OF THE LINES BEGINS WITH A WORD
20. ? WHICH BEGINS WITH P SO THAT
21. ? THE MARKER MUST BE CHANGED.
22. ? INDENTED MATERIAL
23. ? WON'T BE ALIGNED
24. ? ATTN***
COMMAND ? CH 'P' 1 TO '\$' IN ALL NOLIST
COMMAND ? SET LENGTH = 30
COMMAND ? ALIGN 19/L MARKER = \$

COMMAND ? LIST 8/L
8. ALIGNMENT PROCESS, BUT
9. NOTE THAT ONE OF THE LINES
10. BEGINS WITH A WORD WHICH
11. BEGINS WITH P SO THAT THE
12. MARKER MUST BE CHANGED.
13. ? INDENTED MATERIAL
14. ? WON'T BE ALIGNED
COMMAND ? ALIGN ALL MARKER = \$
COMMAND ? LIST
1. \$ THIS IS A SAMPLE DATA SET
2. USED TO SHOW THE EFFECTS OF
3. POSING THE ALIGN COMMAND TO
4. WYLBUR.
5. \$ THE P MARKERS IN COLUMN
6. ONE CAUSE THE RESTARTING OF
7. THE ALIGNMENT PROCESS, BUT
8. NOTE THAT ONE OF THE LINES
9. BEGINS WITH A WORD WHICH
10. BEGINS WITH P SO THAT THE
11. MARKER MUST BE CHANGED.
12. ? INDENTED MATERIAL
13. ? WON'T BE ALIGNED
COMMAND ?

Note that the align process causes a complete renumbering of the entire data set starting at 1.000 and using the current DELTA.

A temporary LENGTH, for use only during an execution of the ALIGN command, may be specified in the command

COMMAND ? ALIGN ALL LENGTH = 40 MARKER = \$

If WYLBUR attempts to ALIGN a line which has no blanks and is longer than the current LENGTH, an image of that line will be printed with the appended error message TOO LONG and execution of the align command will be aborted. The line that was too long will be lost from the data set and no renumbering will be done except for those lines which have already been aligned before the error occurred (and these numbers will not follow any prescribed method of

determination).

MISCELLANEOUS COMMANDS

The user may destroy any of his external data sets by commanding WYLBUR to scratch them

COMMAND ? SCRATCH DUMMY

WYLBUR will remove the data set specified from existence. WYLBUR will not permit the user to scratch any data set except those which belong to him.

The user may ask WYLBUR to show the names of all the external data sets which belong to the user

COMMAND ? SHOW DSNAME
USER DATA SET DIRECTORY
SYS07
DUMMY 11/13/67
DIRECT 08/05/42
SYS04
MANUAL 02/30/68
COMMAND ?

and as indicated, WYLBUR will return the DSNAME and VOLUME attributes of all external data sets which belong to the user.

The user may specify that only the DSNAMEs of his data sets residing on a particular volume are to be found.

COMMAND ? SHOW DSNAME ON SYS04
USER DATA SET DIRECTORY
SYS04
MANUAL 02/30/68
COMMAND ?

The current time of day, in hours, minutes, and seconds, may be obtained by issuing a SHOW TIME command

COMMAND ? SHOW TIME
19:43:15

WHEN FINISHED WITH EDITING A DATA SET

The user will most likely want to save a copy of the data set which he has been using WYLBUR to construct and correct. This is accomplished by directing WYLBUR to save the current version of the working data set with some specified DSNAME

COMMAND ? SAVE DUMMY
FORMAT ?

WYLBUR always requests what format the data set is to be saved in, and the user may preempt this standard request by giving the format in the command

COMMAND ? SAVE DUMMY CARD

If the data set already exists on an external data set storage device, the user will be told and asked if he wishes WYLBUR to scratch the old version of the data set and replace it with the new version currently being saved.

COMMAND ? SAVE DUMMY EDIT
'TOOO.DUMMY' ALREADY EXISTS. DO YOU WANT TO SCRATCH IT?
REPLY ? YES

A YES answer causes WYLBUR to scratch the old and save the new version of the data set. A NO answer causes WYLBUR to terminate the processing of the SAVE command and request a new command from the user. The user may preempt the standard request for permission to scratch an existing data set by appending the phrase SCRATCH to the end of the SAVE command.

A data set may be saved as a private file by specifying

PRIVATE.

COMMAND ? SAVE DUMMY ON SYS04 PRIVATE

Any data set saved in this manner will be available for access only by a user signed on with the correct account number - no other user may access the data set, even by specifying its full DSNAME. The full DSNAME will be XXXX.*.DUMMY, where XXXX is the user's account number.

The user may designate that the data set be saved on a particular storage device by giving its name in the SAVE command

COMMAND ? SAVE DUMMY ON SYS07 CARD

If the user does not specify the desired VOLUME attribute, WYLBUR will put the data set onto any device which has room, and tell the user where the data set was stored.

Normally, line numbers will not be retained when a data set is saved in CARD format. Instead, the data set will be renumbered (from 1.000 by DELTA) when the data set is used again. The user may, however, specify that he wants the line numbers retained

COMMAND ? SAVE DUMMY ON SYS06 CARD NUMBERED

or

COMMAND ? SAVE DUMMY ON SYS06 LRECL=80 NUMBERED

Note that this option is only valid when 80 byte records are being saved. The line numbers will overwrite the contents of character positions 73-80. If the user has saved a data set with line numbers retained and in 80 byte records, he must explicitly tell WYLBUR this fact when he next uses the data set

COMMAND ? USE DUMMY ON SYS06 CARD NUMBERED

If this isn't done, the old, retained line numbers will appear in column positions 73-80 and the data set will be renumbered from 1.000 by DELTA. When NUMBERED is specified, the line numbers for the data set are taken as the contents of column positions 73-80 and these columns are replaced with blank characters. If the contents of character positions 73-80 are not numeric line numbers in ascending order, WYLBUR will do its own line numbering. The numbers must be in ascending order and if two lines appear with the same line number, then only the second will be retained in the working data set.

If the data set is being saved in CARD, PRINT, or LRECL format, a blocking factor may be specified, indicating how many lines of length LRECL should be placed into each logical record. The product of LRECL and the blocking factor must be less than or equal to 3520. The blocking factor is specified as an integer enclosed in parentheses.

COMMAND ? SAVE DUMMY ON SYS17 LRECL=80 (44) SCRATCH

COMMAND ? SAVE DUMMY1 (10) PRINT ON SYS05

COMMAND ?

In the first case above, each block will contain 44 80-byte records. In the second, each block will contain 10 133-byte records.

After the user has saved a copy of the working data set in the desired format, or if he does not wish to keep a copy of the working data set, he may clear the working data

set (i.e. return the working data set to its initial empty state) by commanding WYLBUR to clear

COMMAND ? CLEAR

The user is cautioned against indiscriminate use of this command, since its effect is irreversible (unless, of course, a copy of the data set has been saved). CLEAR only affects the working data set, and the settings of the global parameters are unaffected.

END OF SESSION - LOGOFF PROCEDURE

All the user need do to end the session is issue a LOGOFF command

COMMAND ? LOGOFF

WYLBUR will reply with various statistics on the session

COMMAND ? LOGOFF
ELAPSED TIME = 00:04:32
END OF SESSION
#

The elapsed time is the time which the user has been sitting at the terminal. In addition, WYLBUR keeps track of activity at a terminal and if nothing has been typed in at a logged on terminal for a period of five minutes WYLBUR will ask the user to take some action to show that he is still there

COMMAND ? ARE YOU STILL THERE...

If the user doesn't type something in reply (carriage return is sufficient) then WYLBUR will give him another five minutes and then ask again

COMMAND ? ARE YOU STILL THERE...

COMMAND ? TYPE SOMETHING OR YOU WILL BE LOGGED OFF.

If no reply is made, WYLBUR will logoff the terminal.

The user is warned that the logoff command includes an implicit clear command -- the current working data set is not saved. The user must save the working data set, if he wishes to, before logging off.

GETTING HELP FROM WYLBUR

Frequently, the user may want some information or a review of some command or aspect of WYLBUR. For this purpose, the data set T000.WYLHELP may be called. This data set contains short explanations of the commands plus explanations of the use of ATTN, carriage return, line numbers, etc. The following procedure should be used to obtain help.

COMMAND ? SAVE

COMMAND ? CLEAR

COMMAND ? USE &T000.WYLHELP EDIT

COMMAND ? LIST 'NAME'

This will be a listing of the first line which pertains to information on NAME, which is a command or some other aspect of WYLBUR.

COMMAND ? LIST N/LAST

This listing will be an outline of the information which the user desires. N is the line number of the line listed in response to the LIST 'NAME' command. The user can abort the listing when he has information desired.

COMMAND ?

The file T000.NEWS contains information on changes and additions that have been made to WYLBUR subsequent to the printing of this manual. As in the WYLHELP data set, a directory to the information is present at the beginning of the data set.

APPENDIX B

JCL Card Decks
For Various Tasks

```
C USE THIS JCL TO PUT COMPILER OUTPUT AND GO-STEP OUTPUT
C ON THE SAME DATA SET.  SUBSTITUTE YOUR OWN ACCOUNT
C NUMBER, DSNNAME AND SPACE PARAMETER IN CARD 5.  (1000 IS THE NUMBER OF
C OF LINES (133 BYTES EACH) EXPECTED WITH 15*(100) MORE ALLOWED).
//STEP1 EXEC FORTHCLG
//FORT.SYSPRINT DD SYSOUT=,DISP=(NEW,PASS),
// DCB=(RECFM=F,LRECL=133,BLKSIZE=133),
// VOLUME=SER=SYS04,UNIT=2314,
// SPACE=(133,(1000,100),RLSE),DSNAME=ACCT.DSNAMEXX
//FORT.SYSIN DD *
// YOUR SOURCE DECK
/*
//GO.FT06F001 DD SYSOUT=,DSNAME=*.STEP1.FORT.SYSPRINT,DISP=(MOD,KEEP)
//GO.SYSIN DD *
// YOUR DATA
/*
C -----
C USE THIS JCL TO PUT COMPILER OUTPUT ON A SEPARATE DATA SET.
C MODIFY CARD 4 WITH YOUR OWN ACCOUNT NUMBER, DSNNAME AND
C EXPECTED NUMBER OF LINES OF OUTPUT.
//FORT.SYSPRINT DD SYSOUT=,DISP=(NEW,KEEP),
// DCB=(RECFM=F,LRECL=133,BLKSIZE=133),
// VOLUME=SER=SYS04,UNIT=2314,
// SPACE=(133,(1000,100),RLSE),DSNAME=ACCT.DSNAMEXX
C
```

X
X
X

X
X
X

```

C USE THIS JCL TO PUT LINK-EDIT OUTPUT ON A SEPARATE DATA SET.
C LINK-EDIT OUTPUT MAY NOT BE PUT ON THE SAME DATA SET AS THE
C COMPILER OUTPUT OR THE GO-STEP OUTPUT.
C (MODIFY CARD 3 ACCORDINGLY).
//LKED.SYSPRINT DD SYSOUT=,DISP=(NEW,KEEP),
//              VOLUME=SER=SYS04,UNIT=2314,
//              SPACE=(133,(1000,100),RLSE),DSNAME=ACCT.DSNAMEXX
C
C USE THIS JCL TO PUT GO-STEP OUTPUT ON A SEPARATE DATA SET.
C THIS JCL MAY ALSO BE USED TO PUT BOTH COMPILER AND EXECUTION
C OUTPUT FROM WATFOR ONTO A DATA SET. THE COMPILER AND EXECUTION
C LISTINGS MAY NOT BE SEPARATED UNDER WATFOR.
C (MODIFY THE LAST OF THE 4 CARDS ACCORDING TO YOUR SPACE AND
C DSNAME REQUIREMENTS)
//GO.FT06F001 DD SYSOUT=,DISP=(NEW,KEEP),
//              DCB=(RECFM=FB,LRECL=133,BLKSIZE=3458),
//              VOLUME=SER=SYS04,UNIT=2314,
//              SPACE=(133,(1000,100),RLSE),DSNAME=ACCT.DSNAMEXX
C -----
C
C EXAMPLE:
C AFTER YOU HAVE TYPED YOUR JOBCARD (AS LINE 1), AND YOU WISH TO USE
C 5/10 OF THE FOREGOING, YOU MAY TYPE:
C COMMAND ? COPY 5/10 TO 2 FROM &T000.JCL
C COMMAND ? MOD 6
C 6. //              SPACE=(133,(1000,100),RLSE),DSNAME=ACCT.DSNAMEXX
C ALTERS ?
C NOW MODIFY THE ACCT.DSNAMEXX PARM (AND THE SPACE PARM IF NECESSARY).
C

```

X X

X X X

```

C -----
C USE THIS JCL TO PUT CARD DECKS INTO A DATA SET, TO BE
C REFERENCED LATER FROM A WYLBUR TERMINAL.
//STEP1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT2 DD DISP=(NEW,KEEP),VOLUME=SER=SYS04,UNIT=2314,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3520),
//          SPACE=(TRK,(NN,2),RLSE),DSNAME=ACCT.DSNAMEXX
//SYSIN DD DUMMY
//SYSUT1 DD DATA
//          YOUR CARDS (DO NOT INCLUDE ANY /* CARDS)
/*
C
C PUT YOUR ACCOUNT NUMBER AND DSNAME IN THE ACCT.DSNAMEXX FIELD.
C THIS BLOCKING FACTOR ALLOWS 88 CARDS PER TRACK. SET NN, THE NUMBER
C OF TRACKS TO BE ALLOCATED, ACCORDINGLY.
C -----
C USE THIS JCL TO PUT LISTINGS FROM THE LEVEL F ASSEMBLER ON A DATA
C SET TO BE REFERENCED LATER FROM TERMINALS. THE BLOCKING FACTOR ALLOWS
C APPROXIMATELY 70 LINES PER TRACK ON THE 2314; SET NN, THE NUMBER OF
C TRACKS TO BE ALLOCATED, ACCORDINGLY.
//STEPNAME EXEC ASMFCLG
//ASM.SYSPRINT DD SYSOUT=,VOLUME=SER=SYS14,UNIT=2314,
//          DCB=(RECFM=FB,BLKSIZE=3509,LRECL=121),DISP=(NEW,KEEP),
//          SPACE=(TRK,(NN,2),RLSE),DSNAME=ACCT.DSNAMEXX
//ASM.SYSIN DD *
//          YOUR ASSEMBLER LANGUAGE PROGRAM
/*

```

X X

X X

APPENDIX C

Index and Outline of Commands

This appendix gives a succinct review of all WYLBUR commands as well as references to information in the text of this manual on the commands and other key features of WYLBUR.

For each command, the format is given in outline form under the following rules

Syntactic types are denoted by enclosure in <...> brackets. Each syntactic type is discussed under its own heading.

Optional phrases are denoted by enclosure in [...] brackets.

Alternative choices are separated by a | character and are enclosed in (...) brackets.

VALID ABBREVIATIONS

BY	none	ON	none
SKIP	none	EDIT	none
CARD	none	PRINT	none
LRECL	none	URGENT	U
PRIORITY	P	STANDARD	S
IDLE	I	OVERNITE	O
NOPRINT	N	EXPRESS	X
OVERFLOW	OVER	COLLECT	COL, C
USE	none	SAVE	none
SCRATCH	none	CLEAR	none
RUN	none	LOCATE	none
LOGOFF	none	LIST	none
NUMBER	none	INSERT	INS
REPLACE	REP	COPY	none
MOVE	none	SET	none
SHOW	none	MODIFY	MOD
CHANGE	CH	PUNCH	none
ALIGN	none	FIRST	F
LAST	L	END	none
LENGTH	LEN	UNNUMBERED	UNN

NOLIST	none	NOTEXT	none
OFFLINE	OFF	BIN	B
ACCOUNT	ACCT,A	MARKER	none
TO	none	FROM	none
DELTA	none	TABS	none
UPLOW	none	UPPER	none
OFF	none	CASE	none
OPERATOR	none	TIME	none
STATUS	none	PRINT	none
IN	none	DELETE	DEL
DSNAMES	none	NUMBERED	none
MC	none	PRIVATE	none

ALIGN COMMAND

ALIGN <EXPLICIT RANGE> [LENGTH=XXX] MARKER=C

This causes the contents of the lines in the specified <EXPLICIT RANGE> to be changed so that as many full words as possible are in a line but such that no line contains more than LENGTH characters. Splits in a line are made only at blanks. After the lines in the specified <EXPLICIT RANGE> are aligned, WYLBUR rennumbers the entire data set. One blank is inserted after each word unless it ends with a . or ? or !, which force two blanks. (See page E-59 ff.)

If LENGTH is not specified in the command, the current value of the global LENGTH is used. (See page E-60, E-29.)

The MARKER option must be specified in the command if this feature is desired. When the MARKER character or a blank occurs in column one of any line in the range being aligned, the alignment process is restarted at that line. (See page E-59.)

ASSOCIATIVE RANGE

A group of lines designated by giving a string, an

occurrence of which in any line in the data set will signify that that line containing the string is to be included in the range. The form of an associative range is

'STRING' [N[/M]] [(MA[,MB ... ,MN])] IN <EXPLICIT RANGE>

(See page E-11 ff.)

Either single quote (') or double quote (") marks may be used to delimit the string, but consistency must be maintained. If the same quote mark appears in the string as is used to delimit the string, the internal occurrence must be repeated twice. Blanks are significant in the string, including leading and trailing blanks. (See page E-11, E-12.)

The IN phrase gives an explicit range in which lines containing an instance of the string are to be searched for. (See page E-13.)

The first set of suffix modifiers indicates column positions within which the string must be contained in any given line. If only the first one of the two column positions is specified, the string is constrained to start in that column. (See page E-13, E-14.)

The second set of suffix modifiers can be of two forms

M M/N

and serve to limit the range to a specific subset of lines, namely those whose ordinal positions in the set of all lines containing the string are given by the modifiers. Thus (3) and (3/7) restrict the subset to the third and

third through seventh lines, respectively, which have an occurrence of the string in the specified explicit range. In other words, if a line is found in the specified explicit range which contains an instance of the string, then that line is included in the associative range only if it is the *l*th line containing the instance and the *l* modifier is given. If no modifiers are given, all lines containing the string and in the specified explicit range are included in the associative range. (See page E-13.)

The correct associative range to retrieve blank lines is the null string - '' or '''. Associative ranges consisting of only blanks (e.g. ' ' or " ") will retrieve lines which have the specified number of blanks and also have at least one non-blank character - blank lines will not be retrieved. (See page E-14.)

ATTN KEY

Typed as other than the first character in a line, this signals that WYLBUR should not pay any attention to the line. (See page E-22, E-51.)

Typed as an answer to any prompt, this signals that WYLBUR is to abort execution of the command. (See page E-23, E-43, E-51, E-52.)

Typed while a listing is being done, this key suspends the rest of the listing.

Typed while WYLBUR is putting out a message at the user's terminal, this suspends the printing of the rest of

the message.

Given as an answer to a COMMAND ? prompt, this key puts the user into collect mode. (See page E-21.)

BACK-SPACE KEY

Back-spacing over any character typed by the user erases it from the line. This applies to command as well as text lines. (See page E-21 ff.)

CARRIAGE RETURN

This must be the final character of every line. It signals to WYLBUR that it should look at the line and take appropriate action.

Given as the only alteration to be made to a line (in the MODIFY command), it signals that all modifications have been made to the line. (See page E-50.)

CHANGE COMMAND

This command allows the user to change a specified character string in all lines which are in some specified <RANGE> in his working data set.

CHANGE 'STRINGA' [P[/Q]][(N)] TO 'STRINGB' IN <RANGE>
The specified <RANGE> is searched for lines containing an instance of STRINGA. In each of these lines, the specified occurrence of STRINGA is changed to STRINGB. (See page E-52 ff.)

The P and Q modifiers restrict the instance of STRINGA to be wholly contained in columns P through Q of the line.

If only P is specified, STRINGA is restricted to start in column P of the line. (See page E-55.)

If the N modifier is not given, all occurrences of STRINGA in the line are changed. N=3 would specify the third occurrence in the line. (See page E-53.)

CHANGE X[/Y] TO 'STRINGB' IN <RANGE>

In this alternative form, changes are made to columns X through Y of each line in the <RANGE>. If X and Y are both specified then the characters in positions X through Y are replaced by STRINGB. If only X is specified, STRINGB is inserted before character position X. (See page E-54 ff.)

CLEAR COMMAND

This command scratches the current <WORKING DATA SET>. The settings of global parameters are unaffected.

CLEAR

(See page E-64.)

COLLECT MODE

Mode in which a series of lines are entered into the data set with WYLBUR calculating and prompting successive line numbers.

COLLECT [XXXX.XXX] [BY YYYY.YYY]

(See page E-20 ff.)

Lines are collected into the data set beginning with line XXXX.XXX, and deriving successive line numbers by using a DELTA of YYYY.YYY. If YYYY.YYY isn't specified, the current value of the global DELTA is used. If XXXX.XXX

isn't specified the starting number is DELTA higher than the LAST (number of the last line) of the data set. (See page E-24.)

The COLLECT command will not allow overwriting or interleaving existing lines in the data set. (See page E-25.)

If the OVERFLOW global parameter is set ON, lines containing more than LENGTH characters will be truncated at the last blank occurring before LENGTH characters and the remaining characters will be prompted, with the line number, as the start of the contents of the next line to be COLLECTed. (See page E-25 ff.)

COPY COMMAND

This command allows the user to copy a <RANGE> to another spot in the <WORKING DATA SET>.

```
COPY <RANGE> TO (XXXX.XXX|END) [BY YYYY.YYY]
      [FROM <DSNAME> [ON <VOLUME>]]
```

(See page E-44 ff.)

A copy of the specified <RANGE> is placed in the <WORKING DATA SET>, beginning at a line numbered XXXX.XXX and deriving successive line numbers by using YYYY.YYY

If YYYY.YYY is not specified, the default value is the current working DELTA. If the value of DELTA (specified or default) would cause the copied lines to interleave existing lines in the data set, WYLBUR automatically

calculates a new value of DELTA so that interleaving does not take place. (See page E-46.)

The special line number END is defined as DELTA higher than the current LAST. (See page E-47.)

Existing lines in the data set may not be replaced by using the COPY command.

The <RANGE> is taken from the <WORKING DATA SET> unless the FROM option is present, in which case, the <RANGE> will be taken from the named <EXTERNAL DATA SET>, which must be in EDIT format. (See page E-46.)

There is an anomaly when a COPY is done from an <EXTERNAL DATA SET>. This occurs only when the <RANGE> is specified as FIRST, FIRST/FIRST, LAST, or LAST/LAST -- the first or last line of the <EXTERNAL DATA SET> will not be retrieved correctly.

DELETE COMMAND

This allows the user to delete a <RANGE> in the <WORKING DATA SET>.

DELETE <RANGE>

(See page E-41.)

An alternate form may be used. This consists of giving an <LINE NUMBER> followed immediately by a carriage return as the command. (See page E-43.)

DSNAME

This is the name of a data set. Any identifier is legal as long as it has no more than eight characters and

begins with an alphabetic character. WYLBUR automatically appends the user's account number to the beginning of the DSNAME - e.g. DUMMY becomes T000.DUMMY for a user with account number T000. If a user wants to obtain a copy of another user's data set, he must give the full name preceded by & - e.g. &T000.DUMMY. (See page E-14.)

EXPLICIT RANGE

A group of lines designated by giving the upper and lower limits on the line number.

XXXX.XXX/YYYY.YYY

XXXX.XXX must be less than or equal to YYYY.YYY. If XXXX.XXX = YYYY.YYY, then the range may be specified by giving only the single number. (See page E-10 ff.)

The first line of the <WORKING DATA SET> may be referred to by the implicit reference FIRST. Similarly, LAST will reference the last line. The special explicit range ALL is equivalent to FIRST/LAST. (See page E-10.)

An alternative form of an explicit range is just a list of (ten or fewer) line numbers, e.g.

10.01,20,12,125,199.015

The numbers in the list must be in ascending order. (See page E-11.)

EXTERNAL DATA SET

Any data set external to the current scope of attention of a user. (See page E-16.)

FORMAT

The storage configuration of a data set. EDIT format has line numbers stored with the text in a special form. CARD format has lines stored as 80 byte records (i.e. card images) without line numbers unless the user directly tells WYLBUR to store the line numbers also. PRINT format has lines stored as 133 byte records without line numbers. LRECL=XXX has lines stored as XXX byte records - XXX must lie in the range 1 to 133 - without line numbers unless XXX=80 and the user specifies that line numbers are to be stored. (See page E-15 ff.)

INSERT COMMAND

This command allows the user to insert a line of text into the working data set. Only one line may be inserted -- insertion of more than one line must be done by using the COLLECT command.

INSERT <LINE NUMBER>

This command causes WYLBUR to prompt with the specified <LINE NUMBER>, after which the user should type in the contents of the new line. (See page E-42.)

The specified <LINE NUMBER> may not already exist in the <WORKING DATA SET>.

Alternatively, insertion may be accomplished by giving a command consisting of the <LINE NUMBER>, followed by a single blank character, and the contents of the new line of text. In this alternative form, if the specified line

already exists in the <WORKING DATA SET>, then the old contents will be overwritten -- a replacement is done rather than an insertion. (See page E-43.)

LINE NUMBER

Any number of the form XXXX.XXX between .001 and 9999.999. Leading and terminal blanks may be dropped. The decimal point may be dropped in the case that the number is an integer.

0065.000 = 65.000 = 65. = 65

(See page E-7 ff.)

LIST COMMAND

This command allows the user to obtain a listing of part or all of the <WORKING DATA SET>.

```
LIST [ <RANGE> ] [ UNNUMBERED ] [ NOTEXT ] [ MARKER=C ]
    [ OFFLINE BIN YYY [ (N) ] [ ACCOUNT XXXX ] [ 'TITLE' ]
    [ <RUN PRIORITY> ] [ <PRINT PRIORITY> ] [ MC ] ]
```

(See page E-35 ff.)

If no options are specified, the entire data set is printed. (See page E-35.)

The <RANGE> phrase limits the listing to that of some specified set of lines. (See page E-35.)

The UNNUMBERED option produces a listing containing only the text of the lines in the <RANGE>. (See page E-35, E-37.)

The NOTEXT option suppresses printing of the text and the listing will consist of only the line numbers of the

lines in the specified <RANGE>. (See page E-36.)

The effect of the MARKER option is to cause the listing to be suspended whenever a line is reached which contains the MARKER character, C, in column one. The suspended listing is restarted by giving a CR command, and an ATTN command while the listing is suspended will cause the rest of the listing to be aborted. Note that the MARKER option has no effect on an OFFLINE listing or when UNNUMBERED is not specified. (See page E-37.)

The OFFLINE modifier signals that the user wishes a printing of his data set on the 1403 highspeed printers. The user must tell WYLBUR his bin number, YYY. In addition, he may have the charges incurred put onto Account XXXX and if this isn't specified, the charges will be put onto the Account with which the user logged on. (See page E-38.)

The (N) phrase causes the insertion of N blanks at the left on the 1403 output, thus allowing the user to center his output on the paper. If N=0, the WYLBUR listing routine does not put any carriage control characters in print position one and it is assumed that the data set contains the carriage control characters (be careful that the UNNUMBERED option is specified when N=0 so that line numbers are not picked up as carriage control characters). (See page E-38.)

If the TITLE is specified in an OFFLINE list, it will be printed as the heading for the first page of the output.

The TITLE may not have more than 60 characters. (See page E-38.)

The <RUN PRIORITY> and <PRINT PRIORITY> specify the priority that the user wishes to have attached to these two stages of the processing of his job. (See page E-39 ff.)

The MC option is only valid when an UNNUMBERED OFFLINE (0) list is being done with a data set created by one of the IBM assemblers. (See page E-39.)

LOCATE COMMAND

This command allows the user to inquire about the status of any job which he has put into the job queue through WYLBUR. This may be a job created by use of the RUN command or the LIST OFFLINE command.

LOCATE XXX

LOCATE JOBNAME

XXX is the job number which has been assigned to the user's job. (See page E-57.)

In the second form of the command, the batch stream is searched for all jobs having the specified JOBNAME. The job number of each job found will be printed out and the status of the last one found will be determined and printed for the user. (See page E-57.)

LOGOFF COMMAND

This command allows the user to terminate the session. The <WORKING DATA SET> is not automatically saved -- LOGOFF includes an implicit CLEAR.

LOGOFF

(See page E-65.)

MODIFY COMMAND

The MODIFY command allows the user to alter the contents of each line in a specified <RANGE>. Successive lines are prompted for alteration until the <RANGE> is exhausted.

MODIFY <RANGE>

(See page E-48 ff.)

For each line in the specified <RANGE> the following process is followed. First the image of the line as it now stands is printed out for the user. Then the user is prompted to specify an alteration to be made to the line. Alterations are one of the following

Insert - I

The letter I is typed immediately below the character in the line before which characters are to be inserted. This indicator is followed by the string of characters which are to be inserted. The string of characters to be inserted is terminated by a carriage return. (See page E-49.)

Replace - R

This Indicator is typed immediately below the first character to be replaced in the line. The string of replacement characters is typed immediately following the indicator. As many characters are replaced as there are characters in the string of replacement characters. (See page E-49.)

Delete - D

A string of characters to be deleted from the line is delineated by typing a D under each successive character to be removed. The user need only type a D underneath the first and last character in the string to be deleted -- all characters between the two D's are also deleted. (See page E-48.)

The deletion range may be terminated by typing

either an I or R indicator. If terminated by an R indicator, the characters following the deleted string are replaced by the replacement string that the user specifies. If the deletion range is terminated by an I indicator, the effect is replacement of the deleted characters with the string to be inserted. (See page E-50.)

After the alteration is specified, the user should type a return. WYLBUR will make the indicated alterations in the line and type the new image of the line out for the user to inspect. Then WYLBUR will prompt for more alterations. The user signals that all alterations have been made by typing only a carriage return in answer to the alteration prompt. This will cause WYLBUR to put the new image of the line into the data set and prompt for alterations to the next line in the <RANGE>. (See page E-50.)

The printing of the altered image of the line may be suspended by typing an N indicator into the alteration line before any other alteration indicators. (See page E-51.)

The user may have WYLBUR type out the image, as it currently stands, of the line being modified by typing a few blanks into the alteration line and issuing a carriage return. (See page E-51.)

The user may renege on all alterations made to a line by typing an ATTN as the first character in the alters line. WYLBUR will forget about all alterations which the user may have made to the line presently being altered and the image of the line will remain what it was before alterations were made. (See page E-52.)

MOVE COMMAND

This command moves a specified <RANGE>. It is essentially a COPY command followed by a DELETE command, where the old instances of the copied lines are deleted.

MOVE <RANGE> TO (XXXX.XXX|END) [BY YYYY.YYY]

(See page E-47 ff.)

If YYYY.YYY is not specified, the current value of the global parameter DELTA is used in its place. (See page E-31.)

The special line number END is defined as DELTA higher than the current LAST. (See page E-47.)

The user may not MOVE lines from an <EXTERNAL DATA SET> into the <WORKING DATA SET>. (See page E-47.)

NUMBER COMMAND

This permits the user to have his entire data set renumbered.

NUMBER [XXXX.XXX] [BY YYYY.YYY]

(See page E-41.)

XXX.XXX is the new number for the first line in the data set. If it is not specified, 1.000 is used. (See page E-41.)

YYYY.YYY is the DELTA to be used in deriving successive line numbers. If it is not specified, the current value of the global parameter DELTA is used. (See page E-31.)

PRINT PRIORITY

The priority to be attached to the print phase of a job. Must be one of the following, which are in decreasing order

URGENT
 PRIORITY
 STANDARD
 IDLE
 OVERNITE
 NOPRINT - don't do the print phase

STANDARD is the default priority. (See page E-39 ff.)

PUNCH COMMAND

This command allows the user to produce a punched card deck of a <RANGE> in the <WORKING DATA SET>.

PUNCH <RANGE> [UNNUMBERED] [ACCOUNT XXXX] [BIN YYY]
 [<RUN PRIORITY>] [<PRINT PRIORITY>]

(See page E-39.)

The ACCOUNT and BIN options are the same as in the LIST OFFLINE command. (See page E-38.)

If the UNNUMBERED option isn't specified, the line numbers will be punched into columns 73-80 of the cards. (See page E-38.)

The PUNCH command automatically produces an offline listing of all the cards that are punched unless a <PRINT PRIORITY> of NOPRINT is specified. (See page E-38.)

The <RUN PRIORITY> and <PRINT PRIORITY> specify the priority that the user wishes to have attached to these two stages of the processing of his job. (See page E-39 ff.)

RANGE

This is either an <EXPLICIT RANGE> or an <ASSOCIATIVE RANGE>.

REPLACE COMMAND

This allows the user to replace the content string of any specified <RANGE> in the <WORKING DATA SET>.

REPLACE <RANGE>

(See page E-42 ff.)

WYLBUR will prompt with successive line numbers in the specified <RANGE>, expecting the user to type in the new contents after each prompt. This is continued until the <RANGE> is exhausted. (See page E-43.)

Alternatively, the user may replace the contents of a single line by giving the <LINE NUMBER>, followed by a single blank, and then the new contents of the line. (See alternative form of the INSERT command.) (See page E-43.)

RUN COMMAND

This command allows the user to put the <WORKING DATA SET> into the HASP job input stream of the 360/67

RUN [UNNUMBERED] [<RUN PRIORITY>] [<PRINT PRIORITY>]

(See page E-56 ff.)

The data set will be converted to CARD format (remember that the <WORKING DATA SET> is always in edit format while being worked upon). (See page E-56.)

If the UNNUMBERED option is not given, the line numbers

of the lines of text will be put into columns 73-80. (See page E-56.)

The <RUN PRIORITY> and <PRINT PRIORITY> specify the priority that the user wishes to have attached to these two stages of the processing of his job. (See page E-57.)

RUN PRIORITY

The priority to be attached to the execution phase of a job. Must be one of the following, which are in decreasing order

URGENT
PRIORITY
EXPRESS
STANDARD
IDLE
OVERNITE

STANDARD is the default priority for jobs entered via the card readers and EXPRESS is the default priority for jobs entered via WYLBUR. See the fuller explanation of priorities in the main text of the user's manual. (See page E-39 ff.)

SAVE COMMAND

The SAVE command tells WYLBUR to save the current <WORKING DATA SET>, having a name which the user specifies, on some external storage device.

SAVE <DSNAME> [ON <VOLUME>] [<FORMAT>] [(N)]
[SCRATCH] [NUMBERED] [PRIVATE]

(See page E-61 ff.)

The data set will be saved with the expanded name

XXXX.<DSNAME>, where XXXX is the user's account number.

If the <FORMAT> is not specified, WYLBUR will ask the user what <FORMAT> he desires. (See page E-62.)

The (N) phrase specifies a blocking factor - the number of lines which should be placed into each record. (See page E-64.)

If an old copy of the data set already exists on the storage device, WYLBUR will ask the user whether he wishes the old copy scratched before storing the new copy, unless the SCRATCH option is specified. (See page E-62.)

If no <VOLUME> is given, WYLBUR will store the data set on the first available storage device which it finds. (See page E-63.)

The NUMBERED option is only valid when <FORMAT> is CARD or LRECL = 80 and causes the line numbers to be retained as the contents of character positions 73-80. (See page E-63.)

If the data set is saved with a PRIVATE option, then only a user signed on with the appropriate account number will be able to access the data set. The full DSNAME will be XXXX.*.DSNAME, where XXXX is the user's account number. (See page E-22.)

SCRATCH COMMAND

This command allows the user to scratch any <EXTERNAL DATA SET> which belongs to him.

SCRATCH <DSNAME> [ON <VOLUME>]

(See page E-61.)

SET COMMANDS

SET DELTA = XXXX.XXX

This sets the value of the global DELTA, and hence the default value to be used in all commands where a DELTA may be specified. (See page E-31.)

XXXX.XXX may be any value between .001 and 9999.999 (inclusive).

DELTA has a value of 1.000 unless the user specifies otherwise.

SET TABS [=NA[,NB ... ,NN]]

This allows the user to inform WYLBUR of what tab setting are in effect at the terminals. WYLBUR will prompt with directions telling the user what to do. If no tabs have been set (or if not enough tabs have been set - e.g. only three set and the tab key pressed four or more times) any use of the tab key will cause WYLBUR to forget about processing the line and issue an error message. (See page E-29 ff.)

SET LENGTH = N

The LENGTH is the number of characters which may be in any line of text. If the user types in a line of text containing more than this number of characters, WYLBUR will accept the line but issue a warning message to tell the user that he has exceeded the limit which was specified. (See page E-29.)

N may be an integer between 1 and 133 (inclusive).

LENGTH has a value of 72 unless the user specifies otherwise.

SET UPLow

SET UPPER

These commands govern the case of alphabetic characters. Normally, all alphabetic characters are recognized as upper case. If UPLow is specified, full upper and lower case facilities are available to the user. If UPLow is in effect, the user may revert to the normal case (all alphabetic characters in upper case) by giving the SET UPPER command. (See page E-32 ff.)

SET OVERFLOW = (ON|OFF)

This command turns on and off the OVERFLOW option for COLLECT mode. (See page E-32.)

SHOW COMMANDS

SHOW DELTA
SHOW LENGTH
SHOW TABS
SHOW CASE
SHOW OVERFLOW

These commands allow the user to find out the current value or any of the global parameters. (See page E-33 ff.)

SHOW LAST

LAST is the number of the last line in the <WORKING DATA SET>. (See page E-33.)

SHOW DSNames ON <VOLUME>

This will result in a listing of the <DSNAME>s of all <EXTERNAL DATA SET>s which belong to the user. If the <VOLUME> is specified, the DSNAMEs of data sets belonging to the user and residing only on the specified volume will be retrieved. (See page E-61.)

SHOW OPERATOR 'STRING'

The STRING is sent as a message to the 360 operator. This may be used to communicate set up information. (See page E-57.)

SHOW TIME

The current time, in hours, minutes, and seconds, is printed for the user. (See page E-61.)

SHOW STATUS

The current load of work on the 360, in terms of the jobs being executed and the jobs awaiting processing, is printed out for the user. (See page E-58.)

SHOW RUN

SHOW PRINT

These commands produce a listing of the queues, broken down into the various priority classes, awaiting execution and printing, respectively. (See page E-58.)

SIGN-ON PROCEDURE

This is the dialogue carried on between the user and the Stanford Terminal Processor and is used to determine the validity of the user's account number and the right of the user to employ WYLBUR. (See page E-17 ff.)

USE COMMAND

The USE command tells WYLBUR that the user wants to work on an <EXTERNAL DATA SET> and that WYLBUR is to get a copy of it so that modifications may be made.

USE NAME [ON <VOLUME>] [<FORMAT>] [SKIP XXXX] [NUMBERED]

where NAME is just <DSNAME> for a data set belonging to the user
is &XXXX.<DSNAME> for a data set belonging to a user with Account number XXXX

(See page E-26 ff.)

If the user does not specify the <FORMAT>, WYLBUR will ask that it be specified in answer to a prompt. (See page E-28.)

The SKIP option allows the user to skip over XXXX records at the beginning of the data set being used. The first line of the <WORKING DATA SET> will become the (XXXX+1)th line of the <EXTERNAL DATA SET>. (See page E-29.)

The NUMBERED option is only valid when <FORMAT> is CARD or LRECL = 80 and causes the line numbers to be taken as the contents of character positions 73-80 and these character positions are blanked out. (See page E-63.)

VOLUME

These are names of the storage devices used to hold WYLBUR data sets - e.g. SYS07 or SYS04. See current issues of the User's Bulletin for the names of volumes which are available to the WYLBUR user. (See page E-14.)

WORKING DATA SET

The data set which constitutes the user's scope of attention and upon which the user is affecting changes.

(See page E-16.)

