 5 May 93, 28 Jun 94

Later version may be on office computer in Appleton. This vereion created July 29, 1996]

LONDON STAGE DOCUMENTATION

Here follows documentation of the London Stage files, together with SITAR and SORTSIT, programs especially designed to produce the kinds of searches most useful in research on London Stage files. This documentation was made with WordPerfect and probably requires WP to print it, but you can probably look at it with any word processor. It's in the file LS.DOC in your London Stage setup kit.

Getting Started

load disks into a directory marked "LS."

In order to get up and running, you put the contents of the nine floppy disks marked "LS 1#[-#9"into a new directory of your system, named "LS," and copy them into it using MSBACKUP or MWBACKUP, both of which come with recent versions of MSDOS.) Once you get LS loaded, take a look at a directory listing, and you will see that the LS files are named after the season in which they begin. Thus 1659 is the season of 1659-60. You have to find the next file in sequence to tell where they end. Those files ending in ".npk" are ready to go; the full date has been plugged into every section of every performance entry (see program documentation, below). Those ending in ".lsp" need to go through this process. During the 90's SIU Press was so uninterested, and theatre history so dead, that I turned my attention to other things, thinking, let whoever wants the stuff do the rest of this work: there are nasty little errors that must be fixed, and the entries become so complex that the "xplode" routine may not be able to break them into their proper componenents. The "npk" files take you to 1776. The file TEST.NPK starts off with a number of test cases of possible London Stage syntax that we made up to see if the program could handle them.

You will also see two files with the extension "exe." SITAR does the searches and displays the results for you, and SORTSIT rearranges the output alphabetically. These will run when you command them to. The "code" versions of these programs, the ones that you can edit like a text and also run, but more slowly, are also included. They have the extension ".tru" and run only in conjunction with the programming language TRUE BASIC, which also provides the best way to edit them. You can buy a licensed version of the True BASIC programming system for $99. It is distributed by True Basic, Inc

12 Commerce Avenue
Lebanon, NH 03748
800 872 2742

Asher Research
16 Wellsworth Lane
Rolands Castle
Hampshire, P09 6BY
Tel 0705412668
Fax 0705412668

Sometimes it may be helpful just to read SITAR.TRU to find out what's what, which you can do with any word processor. It consists of four nested loops (input [setup], read, search, sit[ar]out or rep[lace]out) But make a backup copy before you load the program, because a stray keystroke can wreak havoc with performance (just don't ever SAVE the program unless you have made and tested a viable improvement).

LONDON STAGE CODES

To guide the computer and convey information, the source files (".npk" and ".lsp") use the following codes.

* Section header

*p Performance entry and mainpiece

*a Afterpiece section

*s Singing section

*m Music section

*d Dance section

*e Entertainment. (Rope dancing, etc)

*o Opera section (singing with title and parts assigned)

*b Ballet section (dancing with title and parts assigned)

*? Can't remember (music with title and parts assigned)

*? Can't remember (entertainment with title and parts assigned)

$name= Name in parenthetical segment or comment

+title= Title in parenthetical segment or comment

[**parentheical segment**] as delimited by original editors

[**parenthetical segment%** as delimited by LS editors

SITAR

SITAR is a System for Interactive Text Analysis and Retrieval that has some advantages over commercial packages and some disadvantages. Unlike most commercial packages, it is totally independent of the structure of the file it is searching. It is not, for instance, restricted to retrieval of items by line, unless you want it to be. It is sensitive to ordinary features of prose, such as punctuation and paragraphing, as well as to any special features of the text in hand, such as the protocols of a bibliography entry. It enables the user to control the context in which he wishes a match to appear, by choosing where he wants the context to begin and end. It doesn't require any pre-processing of the text file, but you may find that some pre-processing will expedite the work in hand. It has a global search and replace capacity not confined to replacing a letter, word or phrase, but everything within declared boundaries. It will, for instance, replace any sentence containing a given key word or any sentence containing a number. Or, if you wish, it will enable you to edit every string that your search pattern finds.

It can also replace <u>all</u> <u>but</u> certain characters masked in the replacement. It uses six different kinds of "wild cards," characters that will match on a given set of characters, like any number. For instance, if you want to find all the zip codes in an address list, you can specify five number codes in a row.

Its main disadvantage is that it searches sequentially, checking every character from beginning to end until it finds a match. It doesn't work from an index to the text, by which commercial packages go directly to a match instead of searching up to it. Therefore, I suspect it is considerably slower (though not slower than WordPerfect SEARCH); but its loss in speed is a gain in freedom and power. By means of REPLACE, for example, you can automatically number lines, paragraphs, or pages, and then write patterns that will include the numbers in any passage retrieved from the numbered segment.

SITAR retrieves from or replaces in a text file four basic kinds of patterns:

key
key·
·key
key·[string]
[string]·key·[string]

In this representation, "·" stands for the ellipsis, normally written ". . .", which indicates that text intervening between the other elements is being ignored. This dot is invoked by the ESCape key, and by means of it we "escape" the text between the boundary strings and the key. The word "key" designates the character string for which SITAR searches first. If SITAR finds a match to the key, it then supplies the "hit" (matched key) in a context determined by the "[string]s" delimiting the left and right boundaries of the context. Brackets in the representation above indicate that these left and right strings are optional. If no left or no right delimiter is stated, the context on either side is theoretically all the text there is on either side of the key. That much context is just what we have to begin with, so SITAR limits the length of non-delimited ellipses to whatever number of characters the user has declared in his setup parameters.

If I request the pattern

" .·Borges·."

(meaning " ....Borges...."") in a text where Borges is mentioned, I will capture all sentences in which "Borges" appears. If I call for

"Pt·Borges·Pt"

(meaning **"Pt...Borges...Pt"**), where **"Pt"** is the wild character meaning "any punctuation," I will retrieve citations of Borges bounded by the first punctuation on either side of the name. WordPerfect uses two unprinted line feeds to separate paragraphs. With SITAR one can use WP's line feed character to retrieve all paragraphs containing "Borges". The pattern would be

"LF·Borges·LF"

where "LF" is the linefeed character. If I ask for

"·Borges·"

(meaning "**...Borges...**"), I get "Borges" in the middle of a specified number of characters to the left and right, which one can set in advance of a search.

Here follows a table of the wild characters which may be used in search patterns:

ESC displays as **"·"**, and stands for an ellipsis, ". . ."

CTRL/X displays as **"■"** and stands for any regular keyboard character including the space (The basic ASCII set, 127 characters in all, less CTRL characters)

CTRL/O displays as **"Ω"** and stands for the full ASCII character set (255 in all)

CTRL/U displays as "Ü" and stands for any Upper case character

CTRL/L displays as "└" and stands for any Lower case character

CTRL/P displays as "Pt" and stands for any punctuation

CTRL/Q displays as "¥" and stands for any special character (@#%^&*, etc)

CTRL/N displays as "n" and stands for any Number

CTRL/S displays as "ƒ" and stands for a set of characters chosen by the user, including the other wild characters. (A set consisting of "Üpqrs" will match on any Upper case character, or a "p" or a "q" or an "r" or an "s". Can be very useful.

CTRL/R displays as "♪" and stands for a carriage RETURN. Useful in WordPerfect files that use this for a line feed.

CTRL/B displays as "O" and stands for a backspace. Apparently I once had to deal with files that had unprinting backspaces in them.

CTRL/D displays as "δ" and stands for a particular London Stage Date, to be specified by the user.

How it Works

To start the program, type "sitar" at the DOS prompt. When the program comes up it displays the following masthead across the top of the screen:

```
SITAR: Copyright (c) 1987, 1988, 1993, by B R Schneider, Jr, Lawrence University.
Limit [number] Set [characters] Mode [code letters] Infile [filename] Outdir [directory]

To change a setting type 1st letter of parameter:
```

You are now in a position to change any item in the list of settings under the copyright statement:

LIMIT: This variable designates the amount of text that SITAR will display if a pattern match exceeds this number. If, after a hit on the key, the program finds that either the left or right delimiter is farther away than the limit you have designated, the program chops that side of the string hit to fit the limit. If at the settings prompt you type "l", the program responds **"New limit?"** The full screen of a typical monitor screen contains 2000 bytes (25 lines x 80 characters). Since this limit applies to both left and right extensions from the key, if you set "l" to 1000, each hit will be limited to a screenfull. When a left or right context exceeds the specified limit, SITAR chops off the

context at the specified limit and prints a special double bar in the output to indicate that it has done so.

SET: By means of this parameter you can make up your own wild cards, using any of the wild cards already defined plus any list of matching characters you wish to add. When you use the set character "ƒ" (created by CTRL/S) in a key or delimiter string SITAR will match on any member of the set in that position in the string. For example, suppose you wish to capture all entries of an address list that have a ZIP beginning with 54 and have a 7, 8, or 9 in the 3rd digit. At the settings prompt, press "s" and answer the **"New set?"** question with **"789"**. Then write the key of the pattern request thus: **"54ƒ^nn"**. (**"^n"**, CTRL/N, stands for any number.)

MODE: SITAR operates in three basic modes with several options in each. At the settings prompt, when you press "m" the following question pops up:

New mode? (n[ormal], r[eplace], x[plode], s[elect], s[how], or p[rint])

You now have the following options:

"n" produces normal or basic SITAR operation, in which the program searches for matches on the pattern statement and simply displays them on the screen. If you choose this option, SITAR answers

Pattern?

After you answer with a pattern statement like those described above, the program supplies a screenfull of hits and then asks **"More?"** if you answer "y", it gives you the next screenfull. If you answer "n" it stops displaying hits but continues to save them. If you answer by hitting the RETURN (ENTER) key the program cancels the search and returns you to the pattern statement. After completing a search the rpogram prints the number of hits and asks if you wish to save the output. Follow subsequent directions to do so.

"nx": same as above, but unpacks binary dates (only in London Stage files) and displays them in full yr/mo/day at the beginning of each section in the pattern match.

"np": prints output on printer instead of sending it to disk. Chops continuous output into lines shorter than 80 characters. Disregards any formatting codes (prints them, too, if the character is printable).

"nd": Produces a DOS file formatted as above

"r" searches the file just as in normal mode but enables editing of the hits in three ways. When you choose this option the program asks, **"Pattern to be replaced?"** You reply with a pattern statement

as above. It then asks, **"String to replace it with?"**, and here's where the options occur. If you simply want to replace "Henry" with "Charles" every time "Henry" occurs then you write "Henry" in answer to the first question and "Charles" in answer to the second. Unless you type "n" at the "More?" prompt, the replace routine will display all the revised output a screenfull at a time.

To replace only part of a string, mask the part that should remain as is with CTRL/X, which will appear as a solid block in the pattern. Warning: if you replace a pattern with a longer string and the masking character follows the expanded part, a mask won't protect the corresponding part of the search pattern. (Of course, but I found this out the hard way.)

If you wish to edit the hit when you get to it, respond "edxx" to the second question. The program will display the text on the screen and enable you to move the cursor, insert, and delete. Dire will result if the pattern match is longer than one screenfull. If that is so, cancel (CTRL/BREAK), change settings, change delimiters.

"pkxx" at the **"String to replace it with?"** prompt invokes a special London Stage routine that changes all partial dates to full dates, and packs them into two bytes at the head of every section. Use the pattern statement "*·*f*" in which "*f*" stands for the set "Ü." (In most sections the date comes between the beginning of the section and the first letter of the title, the first Upper-case letter. But there are some sections without any caps at all--"*d." and the date should be plugged in between the section code and the period.)

WARNING: patterns with ellipses may cause SITAR to search back beyond the previous hit to find a matching delimiter, thus damaging the file. In this case the program chops off the excessive portion and prints a double bar in the text at that point to let you know it has done so. Since this double bar indicates that you haven't completely replaced the pattern, don't save the output. Or as soon as you see the double bar, cancel (CTRL/BREAK), revise your pattern and try again. It may be that you simply cannot do what you intend to do.

"x" invokes The London Stage "eXplode" routine which makes sort records for every performance of some thing by someone, and every name or title marked in parenthetical text. It comes up with a "Pattern?" request, as in normal mode, and you must state "*·*", which brings in one section at a time for processing.

"xs" Shows each exploded section, one at a time. Use RETURN to pass from one to the next.

"xp" sends exploded output to the Printer.

"s" Selects only those exploded records that contain a given key. The pattern used in this option is "*·KEY·*", in which "KEY" is the word you want to key on.

"ss" Shows each selected hit on the screen.

"sp" prints each selected hit.

INPUT: If you wish to change the input directory and file, press "i", which produces the question, **"New input directory & file?"** and prints the current drive/directory. You may edit this information as you wish.

OUTPUT: When you press "o", the query **"New output directory?"** appears. At this point you can change only the directory. You give the file name after the program has finished and you know what you've got.

When you have made all the settings changes you wish to, press RETURN and SITAR will go on to the next step. These settings are stored in SITAR.FIL when you exit the program. SITAR.FIL must always be in the same directory as SITAR.

When SITAR completes the requested job, it reports the number of hits and asks **"Save output? (y or n)"** RETURN or "n" returns you to the pattern statement without saving the output. "y" results in **"Output file name?"** If the file you choose already exists, you are warned and you can either write over the existing file or name a new one for the output.

TO EXIT:

To cancel a search and go back to the pattern prompt, press the F1 key. To go back to the settings from the pattern prompt press the F1 key. To exit SITAR Press F1 at the settings prompt.

SORTSIT

This program sorts "xploded" SITAR output so as to give alphabetical order to the various components and to digest the evidence in each field category and show it's significance in relation to other categories or items. For example, if you made a file with SITAR of all the performance in which Garrick makes one (use the pattern, "*ƒ·Garrick·*", in which "*ƒ" is a wild set ("pa") that captures both performances of mainpieces (*p") and afterpieces ("*a"), and then sorted it first by actor, then by role, then by title, then by date then by venue then by kind, you would have a very good summary of the people that played with him and a good idea of who played with him most often. For instance, you might get a good idea of which leading lady he preferred. SORTSIT is designed to sort SITAR records in whatever order you prefer. When you run it it asks the following question:

File?

Answer this question with the name of the file. You made this file when you concluded the search you now want to sort. You should have ended it with an ".srt" extension (e.g. "garrick.srt"), but if it happens not to have the right extension, you can of course RENAME the file. Since the program expects the ".srt" extension you write only the main part of the filename, e.g., "garrick". Next, the program asks this question:

Order? (k[ind], d[ate], v[enue], t[itle], r[ole], a[ctor])

In answering, all you do is make a string of the initials of these words. The question gives you the original order. For the order suggested above write

artdvk

and press RETURN (ENTER). When the program has finished sorting it will notify you and the output will be in a file named "Garrick.trs" same as the original but with a reversed extension.