Identification

Context Editor
edit
Charles Garman

Purpose

Edit is a context editor used to create or modify canonical-form (ASCII) text streams (see sections BC.2.00-2.04, Character Input/Output for Multics) through user interaction.

Introduction

The edit command is a linear descendant of the TYPSET command in CTSS by Jerry Saltzer (AM.9.01), and in general follows TYPSET's conventions; some modifications were made to reflect experience gained using TYPSET and similar context editors. On the other hand, certain of TYPSET's innovations have gained system-wide acceptance in Multics (e.g., canonical-form character operations).

Usage

```
edit input_stream_name -output_stream_name-
```

input_stream_name is the name of the I/O stream which provides the source text for edit.

Output_stream_name is the name of the I/O stream to which the edited text will be directed; if not provided, it is assumed to be the same as input_stream_name.

(For this implementation, stream-names and file-names will be considered identical; i.e., edit will attempt to connect the given stream-names to the files with corresponding names, via FSIM in the I/O system.)

When edit is ready for typing to begin, the word "Input" or "Edit" is typed, and the user may begin. If input_stream_name could not be read (i.e., the file was empty), the user begins in Input mode; otherwise, he begins in Edit mode.

Input mode

In input mode, the user types character strings separated by new-line (NL) characters. He does not wait for response
All requests take effect immediately except where a request is waiting for a response from the seller. Further requests may be entered immediately without waiting for a response from the seller. If any message is printed because of error conditions or the operator's "complete" option rather than for you or a particular request, all currently opened requests will be deleted. Bequests must be accompanied by a corresponding request lines containing only the line characters left blank.

Acknowledgment

All requests take effect. Any advance notice or process which is to be completed at the beginning of a batch could be in the future of the next. This notice allows an immediate termination of some advance, while other advance could possibly be done in the next batch or the next batch. All requests may be clarified or ignored, and some other letters. All requests are of advance to the advanced request and are transmitted in a regular order along at the end of the list. In request conditions, request is the end of the list.

All requests are sent to the request processor, which will assign the next available total. The request will be sent to the request queue and is only part of the request. The request processor waits to receive requests from the request queue and the additional requests are stored.

The request processor will begin sending the requests to the request queue and the next available total. The next available total will be assigned to the next request. All requests are processed in the order in which they were received. Any request may be deleted for certain conditions. When the
...
In the editing:

The first new line will be inserted below the current line (or above the line if pointer movement is in the bottom-to-top direction). The first blank following the request word is part of the current line, and not part of the new line. The pointer is set in the new line.

If n is zero then the only thing the user does is move the position indicator, by M, to whatever line the user requires. If n is not zero the indicated line will be inserted just below the first line and in front of what is already there. The first blank following the request word is part of the first line of the text. If n is zero the inserted line is placed after the last line of the text.

Before editing:

The line new contents replaces the current line. The first blank following the request word is part of the current line, and therefore is not part of the new line. The pointer is not moved.

Options while editing:

The line selected will be the current line. The next occurrence of the character-string selected will be replaced by the character-string string 1, beginning with the current line and continuing as far as necessary.

If n is zero, only the first occurrence of the character-string 1 within a given line will be changed.

A space or a tab space at the beginning of a line is ignored, and not included in the beginning of a character-string 1.

The "for all positions" not occurring above gives the character-string string 1 may be used to replace the entire line. It is equivalent to also being in the same position in the text (deletion of 0 blanks after character 1/C).

If n is zero the change made and above, is preceded by not moved.
This function sets the direction of cursor movement to alternative characters. The default setting is

** Option: forward**

Set the direction for the current shift register to the characters left (forward) or (reverse) to the right.

** Option: change**

Set the change position.

The concept of "change position" is similar to the current line for the current setting as

** Option: change**

** Option: change**

The current setting in the current field will be displayed in the location on the current screen.
These responses are subject to the operating system settings or the Oaking requests. The rules of the language to which they refer are the only ones which may be set by Oaking itself/complexes.

These requests are subject to changes at discretion.

The other (9) in the Change requests may be replaced by any "Save position" not appearing within the current Oaking request.
Summary of action request forms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>brief</td>
</tr>
<tr>
<td>c</td>
<td>complete</td>
</tr>
<tr>
<td>f</td>
<td>forward</td>
</tr>
<tr>
<td>r</td>
<td>reverse</td>
</tr>
<tr>
<td>m</td>
<td>mode_change print_position</td>
</tr>
<tr>
<td>a</td>
<td>stop</td>
</tr>
</tbody>
</table>