
EDIT [*line number*] ;

Function:

EDIT is a simple editor providing the capability to perform argument modifications on a TSP command. It is better to use arrow-key editing in DOS/Win TSP, or **Through the Looking Glass** in Windows.

Usage:

Only one argument is allowed with this command, which must be a TSP line number. If this argument is omitted, EDIT will prompt you for modifications to the previous line. This command is generally used for correcting typos, or modifying lists of series, etc... before requesting re-execution of a procedure. RETRY is identical to EDIT, except that execution of the modified command is automatic upon exit. Also, EDIT is treated as a special case in collect mode (it's execution is not suppressed), while RETRY is not.

The editor will first echo the command you wish to modify, then issue the prompt ">>". Responses to the prompt must consist of an editing command followed by appropriate arguments. Any unique abbreviation for an editing command will suffice (including a single letter). Complete arguments must be entered, even if you only wish to replace a single character. Only one modification per edit prompt may be made. Prompting will continue until an EXIT command or carriage return is given in response.

Arguments:

TSP breaks up everything you type (except data) into a string of "arguments". Each series or variable name is an argument, as is each operator. Statement terminators (semi-colons) and item separators (commas and blanks) are not considered arguments. This example has 10 arguments:

```
          GENR GNPN = GNP / ( DELTA + R )  ;
 argument:   1     2    3   4  5 6   7      8  910
```

The editing commands and their arguments are as follows:

EXIT
editing completed (takes no arguments). A simple carriage return will also be interpreted as EXIT.

DELETE arg n
delete the nth occurrence of "arg" in the command.

REPLACE arg1 arg2 n
replace the nth occurance of "arg1" with "arg2" in the command.

INSERT arg1 arg2 n
insert "arg1" after the nth occurance of "arg2" in the command. If there is only one argument provided, it is inserted at the end of the command.

NOTE: "n" is always assumed to be 1 if absent.

EDIT *for interactive mode only*

Restrictions:

1. Subscripts: you will not be able to edit successfully double subscripts, or subscripts with dates.

2. Parentheses: the editor will not successfully find specific parens in commands which contain lags, leads, or subscripts prior to the paren you are specifying.

Examples:

One type of modification you may wish to make is to change the list of options on a command without having to retype the whole thing (particularly if it is lengthy). As a simple example, here is how you might change a static forecast into a dynamic one, as well as request printing and plotting:

5? FORCST (STATIC,DEPVAR = I) IFIT
<no output since the print option is off>
6? EDIT 5
>> INS PRINT
>> REP STATIC DYNAMIC
>> EX
5. FORCST (PRINT,DYNAMIC,DEPVAR = I) IFIT ;

In this case, execution of the modified command would not take place until specifically requested.

Another common modification is to fix a typo. If you have made a typing eror in line 10, for instance, and wish to correct it the sequence might look like this:

10? **INT DP,DP1,LGNP,TIME,C INVR C,G,LM,TIME ;**
<error message because procedure INST is misspelled>
11? **EDIT 10**
>>REPLACE INT INST 1
>>EXIT
10. INST DP,DP1,LGNP,TIME,C INVR C,G,LM,TIME ;
12? **EXEC 10**
10. INST DP,DP1,LGNP,TIME,C INVR C,G,LM,TIME ;

<output from the INST command.>

There are a number of ways that the previous example could be simplified to reduce typing. First of all, commands may be abbreviated. Note that INT was not a valid abbreviation for INST (see Basic Rules). Second, n=1 is the default on the REPLACE command as is carriage return for the EXIT command, so they may be omitted. Also since line 10 is the previous command, it may be omitted fromthe EDIT command. Lastly, statements 11 and 12 may be combined by subsituting the RETRY command for EDIT. Assuming the same error on line 10, the correction would now look like this:

11? **RET**
>> R INT INST
>>
10. INST DP,DP1,LGNP,TIME,C INVR C,G,LM,TIME ;

<output from the INST command.>