This addendum lists those changes and enhancements that have occurred with release 2.0 of the TC Utilities Package. The User Manual remains correct except for the changes and additions noted in this document.

Summary of Changes/Additions

1. More flexible loading and operating procedures.
2. Three new programs added.
3. Data Entry 1 program changes.
4. Data entry of special character provided.
5. Atomize/De-Atomize programs updated for VP/MVP.
6. New TC file conversion subroutines with examples.
Wang's TC Support Utilities 1 Package may be obtained on a Model 2270-type diskette or Model 2210-type mini-diskette. Package numbers for both media are as follows:

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Package Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 2270-type diskette</td>
<td>195-0026-3</td>
</tr>
<tr>
<td>Model 2210-type diskette</td>
<td>195-0026-8</td>
</tr>
</tbody>
</table>
Section 1.2 CPU and Peripheral Requirements

Minimum hardware configuration:

CPU    - 2200-T with 16K memory
        2200VP with 32K memory
        2200MVP with a 16.25K partition

DISK   - Two disk surfaces are required. These may be 2 mini-disk drives, 2 flexible drives or a hard disk drive with any combination of the two.

PRINTER - Any hard copy output device is recommended but not required.
Section 1.3  General Operating Instructions

It is no longer necessary that the system disk be run from the \( R \) drive.

The loading procedure is as follows:

1) Clear memory
2) Mount the system disk at the desired location
3) Load "START" from the desired location
4) Run
5) Enter the address of the system disk when requested.

The following display appears indicating available programs with their special function key assignments.

TELECOMMUNICATIONS SUPPORT SYSTEM

<table>
<thead>
<tr>
<th>FN KEY</th>
<th>PROGRAM NAME</th>
<th>FN KEY</th>
<th>PROGRAM NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>DATA ENTRY I</td>
<td>03</td>
<td>OFF-LINE PRINT</td>
</tr>
<tr>
<td>01</td>
<td>ATOMIZE</td>
<td>04</td>
<td>TC SUBROUTINES</td>
</tr>
<tr>
<td>02</td>
<td>DE-ATOMIZE</td>
<td>05</td>
<td>LIST/DUMP TC FILES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15</td>
<td>RETURN TO ISS</td>
</tr>
</tbody>
</table>

STOP
The Initialize Atom Table has been removed from the TC Utilities Package. Making a back-up copy of the system disk precludes the need for this program.

Three additional programs have been added to the system disk. They are Off-line Print, TC Subroutines and List/Dump TC Files.

LIST/DUMP TC FILES - This utility allows the printing of disk files in TC file format. Options include CRT or printer and ASCII or hex. When dumped to the CRT, the block, record count and record length are displayed. The utility is quite helpful for verifying that a file is really in TC file format after conversions. Operation is straightforward and self documenting.

OFFLINE PRINT UTILITY - This program is used to recover files which were received via Wang telecommunication emulators using the print spooled to disk option. The program has the ability to specify a list of files on various disk surfaces to be printed. Optional print to the CRT is available. Note that a spooled print file is different from a standard TC file in that a print file has a prefix character on each record for print control (i.e. top of form, skip to channel, etc.). This control byte is actually the ESC sequence received off the line and not a ANSI print control byte. Operation is straightforward and self documenting.

TC SUBROUTINES - Three subroutines ('130-'132) are provided. From the main menu, the routines are loaded allowing the user to save them, modify or build around them. No return to the main menu is provided. Section 4.0 describes their usage for converting, reading or writing of TC files.
Section 2.1 Data Entry I Program

Change to Data Entry I Program are as follows:

1. The edit work area (data file named "EDITAREA") is provided on the system disk. When loading the data entry module it will ask if this file is large enough. If not, the operator will be given the opportunity to select a different address and to create a new edit area up to 1000 sectors in size. If the system disk is designated, the edit area size is fixed as follows:

   | diskette | 729 sectors | 80 column records | 2,181 records maximum |
   |          | 128 column records | 1,454 records maximum |

   | mini-diskette | 50 sectors | 80 column records | 150 records maximum |
   |              | 128 column records | 100 records maximum |

   The only restriction regarding the edit area is that when specifying the system disk address the system platter must be used.

2. TC File Operations and Edit Area Operations are now entered by pressing special function keys. The available operations and their corresponding special function keys are displayed at the top of the CRT. Parameters required to further define the operations are entered by responding to CRT prompts after an operation has been selected.

3. Three functions previously available have been removed. They are the HELP function and the EDIT and INSERT according to the original numbering scheme. (EDIT 0 and INS 0).

4. For data entry operations the explicit column indicator has been replaced by a static column mask. This requires less CPU time to update column positions and therefore will provide for a significant increase in speed of data entry.

5. The "LOCATE" operator under Edit Area Operations has been renamed to "SEARCH". When using the search operator if a comma is to appear in the character string, it must be entered by pressing S.F. '16. For some CRT's this will display as an underline comma; the underlining is ignored for the SEARCH operation.
6. The SEARCH and LIST operations allow output to either the CRT or the printer. The LIST operation provides for output in either normal print format or HEX code format.

7. All record-display level operations continue to work as before. Note that SIN(, COS(, and EXP( serve to enter special characters [, ], /, and → during data entry. However, on a 2200VP or 2200MVP, these keys will not work properly for the SEARCH operations. For 2200VP and MVP systems use the special function keys as indicated:

```
[       S.F. '17
]       S.F. '18
/       S.F. '19
<       S.F. '20
```

The ability to enter special characters in addition to those listed above has been added. Activating this feature requires modifying program module DEl.020A, lines 1220 through 1300 on the system disk. The HEX codes in both the DEFFN' and in the assignment statements can be changed from HEX(20), the current value, to the HEX code of the desired special character. This will enable data entry of special characters by activating special function keys '21 through '25.
Section 3.1 Atomize and De-Atomize Programs

Atomize and De-Atomize programs have been updated to accommodate the 2200VP/2200MVP BASIC text atoms and syntax.

The following is a list of conditions that should be noted in addition to those on page 20.

1. "RE" (used in the cassette statement DATA RESAVE) is not atomized.

2. "ERROR" as used in the VP syntax is not atomized due to conflict with the ON ERROR syntax.

3. The # character used with "#TERM" and "#PART" is not atomized.
Section 4.0 USING THE TC FILE FORMAT SUBROUTINES

TC file format is a disk format used for storing variable length records for subsequent transmission using one of Wang's emulation packages. Contained in the Telecommunication Support Utilities, is a set of subroutines which allow the reading and writing of records in TC file format. These subroutines when used with a user written program allow easy conversion of data file to and from TC file format.

The following variables are defined and used by the subroutines:

- X3, X5, X6, X7, X8 - Reserved
- X1$( ), X2$( ), X4$ - Reserved
- X$( ) - Input Output user variable
- L - Record length
- Q$ - Return Code

The file used must reside on the disk surface selected for #2.

OPEN

The OPEN routine sets-up the required variables and opens or validates the file name. Only one file may be opened at a time. Another open call essentially cancels the previous one.

Call: GOSUB '130 (a,"filename",n)

Where: a = 1 : to open a file for input
       2 : to open a file for output

"filename" - is the filename of a new or old file on disk.

n = Ø : if the filename already exists.
       = value: Equal to the number of sectors to open a new file for. Useful only when opening a file for output.

The variable Q$ is set to one of the following:

- Q$ = " " if Open successful
- = "X" if illegal call

-9-
READ

The read routine copies the next sequential record into X$() and sets L equal to the record length. Note that X$(1) only contains the first 64 bytes of the record with X$(2), X$(3) and X$(4) containing the remainder, up through the record length L. If the last record is encountered Q$ is set to "E".

Call: GOSUB '131
Where: Record is returned in X$()
       Record length is set in L.

The variable Q$ is set to one of the following:

Q$ = "E" if record returned is last record
    " " if record was passed successfully
    "X" if illegal call

WRITE

The write routine adds the record passed in X$() to the opened file. If indicated, an end-of-file is written and the file closed.

Call: GOSUB '132 (1,n)
Where: l = record length
       n = l if last record
            0 if not last record

X$() contains the record length of l

Upon return, the variable Q$ is set to one of the following:

Q$ = " " if successfully written
    "X" if illegal call

Only errors in the calling sequence are trapped. Disk errors including file not found or insufficient disk space will terminate with BASIC errors.
The following sample programs demonstrate the use of the routines. Note that lines 20 and 9000–9039 are used for the subroutines.

To list out a TC file:

```
100 SELECT #2 350
120 INPUT "ENTER FILENAME",N$
130 GOSUB '130(1,N$,0)
135 IF G$="X" THEN 250
140 GOSUB '131
150 IF G$="X" THEN 260
160 IF L=0 THEN 180
170 $GIO/005(A000 400D 400A,Z$) X$(<1,L>
180 IF G$="E" THEN 140
190 STOP "LISTING DONE"
250 STOP "OPEN ERROR"
260 STOP "READ ERROR"
```

Or, without using $GIO:

```
100 SELECT #2 350
120 INPUT "ENTER FILENAME",N$
130 GOSUB '130(1,N$,0)
135 IF G$="X" THEN 270
140 GOSUB '131
150 IF G$="X" THEN 280
160 IF L=0 THEN 250
165 I=0
170 IF L<=64 THEN 230
180 FOR I=1 TO INT((L-1)/64)
190 PRINT X$(I);
200 L=L-64
210 NEXT I
220 IF L=0 THEN 240
230 PRINT STR$(X$(I+1),1,L);
240 PRINT
250 IF G$="E" THEN 140
260 STOP "LISTING DONE"
270 STOP "OPEN ERROR"
280 STOP "READ ERROR"
```
Simple Data Entry program to create a new file "TEST"

100 SELECT #2 310
110 GOSUB '130(2,"TEST",20)
115 INIT(20)X$(1)
120 INPUT X$(1)
130 INPUT X$(2)
140 IF STR(X$(1),1,3)="EOF" THEN 170
150 GOSUB '132(80,0)
160 GOTO 115
170 GOSUB '132(3,1)
180 STOP "END INPUT"

The following program converts a KFAM file on 310 to a
TC file on B10. The TC file will be a new file with 20 sectors
allocated and a fixed record length of 120 bytes. (Note: Listing
does not include KFAM nor TC variables and subroutines.)

70 DIM K$(3)64
80 SELECT #2 B10,#3 310,#4 310
90 S2=1
100 GOSUB '130(2,"NEWFILE",20): REM OPEN NEW TC FILE=20 SECTORS
110 IF Q$="X" THEN 993
120 GOSUB '230(1,4,3,1,"EMPLF101",9,"","310","310"): REM OPEN KFAM FILE
130 IF Q$<>" " THEN 998
140 REM LOOP READING KFAM, WRITING TC UNTIL EOF
150 GOSUB '235(1,1): REM READ FIRST
160 IF Q$="E" THEN 300
170 IF Q$<>" " THEN 998
180 DATA LOAD DC #3,K$(0)
190 MAT COPY K$(0)<1,120> TO X$(0)<1,120>
200 GOSUB '132(120,0): REM TC WRITE
210 IF Q$="X" THEN 999
220 GOSUB '237(1,1): REM KFAM READ NEXT
230 GOTO 160
240 GOSUB '132(0,1): REM WRITE TC EOF
310 STOP "CONVERSION DONE"
998 STOP " ERROR IN KFAM CALL"
999 STOP " ERROR IN TC CALL"